



ElAnis Platform - Comprehensive Project Planning

Table of Contents

1. Project Overview
 2. Sprint Planning (10 Sprints)
 3. Work Breakdown Structure (WBS)
 4. Resource Allocation
 5. Gantt Chart
 6. Risk Management Plan
 7. Quality Assurance Plan
-

Project Overview

Project Name: ElAnis - Elderly Care Service Platform

Project Duration: 10 Weeks (MVP)

Project Manager: [Name]

Start Date: [Week 1 Start Date]

Target Launch: [Week 10 End Date]

Methodology: Agile Scrum

- **Sprint Duration:** 1 week
 - **Total Sprints:** 10
 - **Daily Standups:** Yes (15 minutes)
 - **Sprint Reviews:** Every Friday
 - **Sprint Retrospectives:** Every Friday (after review)
-

Sprint Planning (10 Sprints)

Sprint 0: Project Kickoff & Setup (Week 1)

Sprint Goals:

- Initialize project infrastructure
- Set up development environment
- Finalize requirements
- Create initial backlog

Tasks:

Day 1-2: Project Setup

☐ **Project Kickoff Meeting**

- Team introductions
- Role assignments
- Communication channels (Slack, Jira)
- Review project scope and timeline

☐ **Development Environment Setup**

- Install Visual Studio 2022
- Install SQL Server 2022
- Install Node.js 18+
- Setup Git repository (GitHub)
- Configure development branches (main, develop, feature/*)

☐ **Create Project Structure**

- Backend: ASP.NET Core Web API solution
 - ElAnis.API
 - ElAnis.DataAccess
 - ElAnis.Entities
 - ElAnis.Utilities
- Frontend: React + TypeScript project
- Database: SQL Server database creation

Day 3-4: Requirements & Documentation

☐ **Finalize Requirements Document**

- Functional requirements (50+ pages)
- Non-functional requirements

- User stories (50+ stories)

- Acceptance criteria

☐ **Create Database Design**

- Entity Relationship Diagram (ERD)
- Database schema design
- Relationships and constraints
- Indexing strategy

☐ **API Specification**

- OpenAPI/Swagger documentation template
- Endpoint definitions (50+ endpoints)
- Request/response schemas
- Authentication flow

Day 5: Design Kickoff

☐ **UI/UX Design**

- Create design system
- Define color palette
- Typography selection
- Component library structure
- Wireframes for key pages (10+ pages)

Deliverables:

- ☒ Project repository setup
- ☒ Requirements document (final)
- ☒ Database ERD
- ☒ API specification draft
- ☒ Wireframes (low-fidelity)

Story Points: 21 **Team Capacity:** 40 hours/person

Sprint 1: Authentication & User Management (Week 2)

Sprint Goals:

- Implement user registration and login
- Setup JWT authentication
- Create user profile management
- Email verification with OTP

Backend Tasks:

Authentication Module

☐ User Registration (8 points)

POST /api/auth/register

- Email validation
- Password hashing (bcrypt)
- OTP generation
- Send verification email

☐ Email OTP Verification (5 points)

POST /api/auth/verify-otp

- Validate OTP
- Mark email as verified
- Generate JWT token

☐ User Login (5 points)

POST /api/auth/login

- Validate credentials
- Check email verified
- Generate JWT + Refresh Token
- Return user profile

☐ Password Reset (5 points)

POST /api/auth/forgot-password

POST /api/auth/reset-password

- Email reset link
- Validate reset token
- Update password

☐ **Refresh Token (3 points)**

POST /api/auth/refresh-token

- Validate refresh token
- Issue new access token

User Profile Module

☐ **Get User Profile (2 points)**

GET /api/users/profile

- Return current user data

☐ **Update User Profile (3 points)**

PUT /api/users/profile

- Update name, phone, address
- Validation

Frontend Tasks:

☐ **Registration Page (5 points)**

- Form with validation
- Email, password, name, phone
- Terms acceptance checkbox
- Submit and error handling

☐ **OTP Verification Page (3 points)**

- 6-digit OTP input
- Resend OTP button
- Countdown timer

☐ **Login Page (3 points)**

- Email/password form
- "Remember me" checkbox
- Forgot password link
- Social login buttons (future)

☐ **Forgot Password Flow** (3 points)

- Email input page
- Reset password page
- Success confirmation

☐ **User Profile Page** (5 points)

- Display user info
- Edit profile form
- Change password
- Upload profile picture

Testing:

- ☐ Unit tests for auth endpoints (5 points)
- ☐ Integration tests (3 points)
- ☐ Frontend component tests (3 points)

Deliverables:

- ☒ Working authentication system
- ☒ User registration and login
- ☒ Email verification
- ☒ Password reset
- ☒ User profile management
- ☒ 80%+ test coverage

Story Points: 55 **Team Capacity:** 160 hours total

Sprint 2: Provider Application Module (Week 3)

Sprint Goals:

- Provider application submission
- Document upload functionality
- Admin application review workflow
- Email notifications

Backend Tasks:

Provider Application

☐ Submit Application (8 points)

POST /api/providers/apply

- Personal information
- Professional details
- Experience, bio, hourly rate
- Status: Pending

☐ Upload Documents (8 points)

POST /api/providers/upload-document

- National ID (front & back)
- Certificates
- CV/Resume
- File validation (type, size)
- Store in Azure Blob / Local

☐ Get Application Status (2 points)

GET /api/providers/application/{userId}

- Return application status
- Show rejection reason if rejected

Admin Application Review

☐ Get Pending Applications (3 points)

GET /api/admin/applications?status=Pending

- Paginated list
- Filter by status, date

☐ **View Application Details** (2 points)

GET /api/admin/applications/{id}

- Full application data
- Display uploaded documents

☐ **Approve Application** (5 points)

PUT /api/admin/applications/{id}/approve

- Create ServiceProviderProfile
- Set status: Approved
- Send approval email
- Link to User role update

☐ **Reject Application** (3 points)

PUT /api/admin/applications/{id}/reject

- Set status: Rejected
- Store rejection reason
- Send rejection email

Notifications

☐ **Email Service Integration** (5 points)

- Setup SendGrid / SMTP
- Email templates (approval, rejection)
- Send email function

Frontend Tasks:

☐ **Provider Application Form** (13 points)

- Multi-step form (3 steps)
- Step 1: Personal info
- Step 2: Professional details
- Step 3: Document uploads
- Form validation
- Progress indicator

☐ **Document Upload UI** (5 points)

- Drag-and-drop upload
- File preview
- Upload progress bar
- File size/type validation

☐ **Application Status Page** (3 points)

- Show current status
- Display rejection reason
- Resubmit option

☐ **Admin Application List** (5 points)

- Table view with filters
- Search functionality
- Status badges
- View details button

☐ **Admin Application Review** (8 points)

- Display all application data
- View uploaded documents
- Approve/Reject buttons
- Rejection reason modal

Testing:

☐ Backend tests (5 points)

☐ Frontend tests (3 points)

☐ File upload tests (3 points)

Deliverables:

- ☒ Provider application submission
- ☒ Document upload system
- ☒ Admin review workflow
- ☒ Email notifications

Story Points: 76 **Team Capacity:** 160 hours

Sprint 3: Categories & Pricing Management (Week 4)

Sprint Goals:

- Category CRUD operations
- Service pricing management
- Shift-based pricing system
- Admin pricing dashboard

Backend Tasks:

Category Management

☐ Create Category (3 points)

POST /api/admin/categories

- Name, description, icon
- Display order
- Active status

☐ Get All Categories (2 points)

GET /api/categories

- Public endpoint
- Return active categories

☐ Update Category (2 points)

PUT /api/admin/categories/{id}

- Update category details

☐ Delete Category (2 points)

DELETE /api/admin/categories/{id}

- Soft delete (set IsActive = false)

Service Pricing Management

☐ Create Pricing (5 points)

POST /api/admin/service-pricing

- CategoryId, ShiftType
- PricePerShift
- Description
- Prevent duplicates

☐ **Bulk Create Pricing** (8 points)

POST /api/admin/service-pricing/bulk

- Create multiple pricing records
- Category + all shift types
- Validation

☐ **Get Pricing by Category** (3 points)

GET /api/service-pricing/category/{categoryId}

- Return all shift prices
- Public endpoint

☐ **Update Pricing** (3 points)

PUT /api/admin/service-pricing/{id}

- Update price
- Store updated by user
- Track update timestamp

☐ **Get All Categories with Pricing** (5 points)

GET /api/service-pricing/categories-with-pricing

- Return all categories
- Include pricing for each
- Public endpoint

Frontend Tasks:

☐ **Category Management Page** (8 points)

- List all categories
- Add new category modal
- Edit category inline

- Delete confirmation

☐ **Pricing Management Page** (13 points)

- Category selector
- Pricing grid (category × shift)
- Inline editing
- Bulk update capability
- Price history view

☐ **Category Display (Public)** (3 points)

- Category cards on homepage
- Icons and descriptions
- Click to filter providers

Testing:

☐ Backend tests (5 points)

☐ Frontend tests (3 points)

Deliverables:

- ☒ Category management system
- ☒ Service pricing system
- ☒ Bulk pricing operations
- ☒ Admin pricing dashboard

Story Points: 65 **Team Capacity:** 160 hours

Sprint 4: Provider Profile & Availability (Week 5)

Sprint Goals:

- Provider profile management
- Working areas configuration
- Availability calendar system
- Provider dashboard

Backend Tasks:

Provider Profile

☐ Get Provider Profile (3 points)

GET /api/providers/{id}

- Public endpoint
- Return profile with categories
- Include working areas
- Include availability (next 30 days)
- Include shift prices

☐ Update Provider Profile (5 points)

PUT /api/providers/profile

- Update bio, experience
- Update hourly rate
- Upload profile picture

Working Areas

☐ Add Working Area (3 points)

POST /api/providers/working-areas

- Governorate, city, district
- Multiple areas support

☐ Get Working Areas (2 points)

GET /api/providers/{id}/working-areas

- Return all active areas

☐ Update Working Area (2 points)

PUT /api/providers/working-areas/{id}

- Update area details

☐ Delete Working Area (2 points)

DELETE /api/providers/working-areas/{id}

- Soft delete

Provider Availability

☐ Set Availability (8 points)

POST /api/providers/availability

- Date, shift type
- Is available flag
- Bulk update (multiple dates)

☐ Get Availability (3 points)

GET /api/providers/{id}/availability

?start=YYYY-MM-DD&end=YYYY-MM-DD

- Return availability range

☐ Update Availability (3 points)

PUT /api/providers/availability/{id}

- Toggle availability
- Change shift type

Provider Categories

☐ Assign Categories (3 points)

POST /api/providers/categories

- CategoryId array
- Link provider to categories

Frontend Tasks:

☐ Provider Profile Page (Public) (13 points)

- Header with photo, name, rating
- About section (bio, experience)
- Categories badges

- Working areas list
- Pricing table (category × shift)
- Availability calendar (30 days)
- Reviews section
- "Request Service" button

☐ **Provider Dashboard** (8 points)

- Statistics cards:
 - Completed jobs
 - Total earnings
 - Average rating
 - Pending requests
- Quick actions
- Recent bookings

☐ **Provider Profile Settings** (8 points)

- Edit bio, experience
- Upload profile picture
- Update hourly rate
- Select categories

☐ **Working Areas Management** (5 points)

- Add new area form
- List existing areas
- Edit/delete areas
- Governorate/city dropdowns

☐ **Availability Calendar** (13 points)

- Month view calendar
- Click date to toggle availability
- Select shift type dropdown
- Color-coded availability:
 - Green: Available
 - Red: Not available

- Gray: Past dates
- Bulk select (week, month)
- Save changes

Testing:

- ☐ Backend tests (5 points)
- ☐ Frontend tests (5 points)

Deliverables:

- ☒ Provider profile system
- ☒ Working areas management
- ☒ Availability calendar
- ☒ Provider dashboard

Story Points: 89 **Team Capacity:** 160 hours

Sprint 5: Provider Discovery & Search (Week 6)

Sprint Goals:

- Provider search and filtering
- Advanced search functionality
- Provider listing page
- Search optimization

Backend Tasks:

Provider Search

- ☐ **Search Providers** (13 points)


```
GET /api/providers
?available=true
&governorate=Cairo
&city=Nasr City
&categoryId=guid
&search=name
&page=1&pageSize=10
```

- Filter by:
 - Availability status
 - Location (governorate, city)
 - Category
 - Search term (name, bio, areas, categories)
- Pagination
- Sorting (rating, price, name)

☐ **Get Provider Detail** (already done in Sprint 4)

☐ **Search Optimization** (5 points)

- Database indexing
- Query optimization
- Caching strategy

Frontend Tasks:

☐ **Provider Search Page** (21 points)

- **Filters Sidebar:**
 - Availability toggle
 - Governorate dropdown
 - City dropdown (dynamic based on governorate)
 - Category multi-select
 - Rating filter (min stars)
 - Price range slider
- **Search Bar:**
 - Text search
 - Search suggestions
 - Clear button

- **Results Grid:**

- Provider cards
- Pagination
- Loading skeletons
- Empty state

- **Sorting:**

- Sort dropdown (Rating, Price, Name)
- Sort direction toggle

☐ **Provider Card Component** (5 points)

- Profile picture
- Name, rating (stars)
- Categories (badges)
- Location
- "Available Now" indicator
- Hourly rate
- "View Profile" button

☐ **Search Performance** (3 points)

- Debounced search
- Lazy loading
- Infinite scroll (optional)

Testing:

☐ Search functionality tests (5 points)

☐ Performance tests (3 points)

Deliverables:

- ☒ Provider search and filtering
- ☒ Advanced search UI
- ☒ Provider listing page
- ☒ Search optimization

Story Points: 55 **Team Capacity:** 160 hours

Sprint 6: Service Request Module (Week 7)

Sprint Goals:

- Service request creation
- Request workflow (pending → accepted/rejected)
- Provider request management
- Notification system

Backend Tasks:

Service Request Creation

☐ Create Service Request (13 points)

POST /api/requests

- CategoryId, ProviderId
- ShiftType, PreferredDate
- Address, Governorate
- Description

Validation:

- Server-side price calculation
- Provider availability check
- Duplicate request prevention
- Create notification for provider

☐ Get User Requests (3 points)

GET /api/requests/user

- Return user's requests
- Include provider, category details
- Order by date (newest first)

☐ Get Provider Requests (3 points)

GET /api/requests/provider/{providerId}

?status=Pending

- Return provider's requests
- Filter by status
- Include user, category details

☐ **Get Request Details** (2 points)

GET /api/requests/{id}

- Full request details
- Authorization check

Provider Response

☐ **Accept Request** (8 points)

PUT /api/requests/{id}/response

Body: { status: "Accepted" }

- Update status: Accepted
- Set AcceptedAt timestamp
- Create notification for user
- Email notification

☐ **Reject Request** (5 points)

PUT /api/requests/{id}/response

Body: { status: "Rejected", reason: "..." }

- Update status: Rejected
- Store rejection reason
- Create notification for user
- Email notification

Notification System

☐ **Create Notification** (3 points)

POST /api/notifications

- UserId, Title, Message
- Type, ServiceRequestId

☐ **Get User Notifications** (3 points)

GET /api/notifications

?unreadOnly=true

- Return user's notifications
- Mark as read option

☐ **Get Unread Count** (2 points)

GET /api/notifications/unread-count

- Return count of unread

☐ **Mark as Read** (2 points)

PUT /api/notifications/{id}/read

- Mark notification as read

Frontend Tasks:

☐ **Booking Modal/Page** (13 points)

- **Step 1:** Select category (from provider)
- **Step 2:** Select shift type (show price)
- **Step 3:** Select date (from available dates)
- **Step 4:** Enter address
- **Step 5:** Add notes (optional)
- **Review:** Show summary
- Submit button
- Loading state
- Success/error messages

☐ **User Requests Page** (13 points)

- List all user requests

- Filter by status tabs
- Request card:
 - Provider info
 - Service details
 - Status badge
 - Date, time
 - Price
 - View details button
 - Pay button (if accepted)

☐ **Provider Requests Page** (13 points)

- List incoming requests
- Filter by status
- Request card:
 - User info
 - Service details
 - Date, time, location
 - Price
 - Accept/Reject buttons
- Accept confirmation modal
- Reject modal with reason

☐ **Notification Center** (8 points)

- Bell icon with unread count
- Dropdown notification list
- Mark as read on click
- Link to related request
- "View All" link
- Empty state

Testing:

☐ **Backend tests** (8 points)

☐ Frontend tests (5 points)

Deliverables:

- ☒ Service request creation
- ☒ Request workflow
- ☒ Provider response system
- ☒ Notification system

Story Points: 104 **Team Capacity:** 160 hours

Sprint 7: Payment Integration (Week 8)

Sprint Goals:

- Payment gateway integration (Paymob)
- Payment processing flow
- Payment status tracking
- Refund mechanism

Backend Tasks:

Payment Integration

☐ **Paymob Integration** (13 points)

- Setup Paymob account
- Generate API keys
- Create payment intent
- Handle payment callbacks
- Verify payment signature

☐ **Create Payment** (8 points)

POST /api/payments

- ServiceRequestId
- Validate request status (must be Accepted)
- Validate amount (server-side)
- Create Payment entity
- Redirect to payment gateway

☐ **Payment Webhook** (8 points)

POST /api/payments/webhook

- Receive payment callback
- Verify signature
- Update payment status
- Update request status (Paid)
- Send confirmation emails

☐ **Get Payment Status** (2 points)

GET /api/payments/{requestId}

- Return payment status

☐ **Refund Payment** (5 points)

POST /api/admin/payments/{id}/refund

- Admin only
- Process refund via gateway
- Update payment status
- Send refund confirmation

Frontend Tasks:

☐ **Payment Page** (13 points)

- Display request summary
- Show payment amount
- Payment method selection:
 - Credit/Debit card
 - Vodafone Cash
 - Fawry

- Paymob iframe integration
- Terms and conditions checkbox
- "Pay Now" button
- Loading state

☐ **Payment Success Page** (3 points)

- Success message
- Receipt details
- Download receipt button
- "View Request" link
- "Book Another" button

☐ **Payment Failed Page** (3 points)

- Error message
- Retry payment button
- Contact support link

☐ **Payment Status in Requests** (5 points)

- Show payment status badge
- Display "Pay Now" button if accepted
- Show receipt link if paid

Testing:

- ☐ Payment integration tests (8 points)
- ☐ Webhook tests (5 points)
- ☐ Frontend payment flow tests (3 points)

Deliverables:

- ☒ Payment gateway integration
- ☒ Payment processing
- ☒ Payment webhook handling
- ☒ Refund system

Story Points: 76 **Team Capacity:** 160 hours

Sprint 8: Reviews & Rating System (Week 9)

Sprint Goals:

- Review submission after service
- Rating calculation
- Review display on provider profile
- Review moderation (admin)

Backend Tasks:

Review System

☐ Submit Review (8 points)

POST /api/reviews

- ServiceRequestId
- Rating (1-5 stars)
- Comment (optional)
- Validate: request completed
- Validate: one review per request
- Update provider average rating
- Update provider total reviews

☐ Get Provider Reviews (3 points)

GET /api/reviews/provider/{providerId}

?page=1&pageSize=10

- Return paginated reviews
- Order by date (newest first)

☐ Get User Reviews (2 points)

GET /api/reviews/user

- Return user's submitted reviews

☐ Calculate Average Rating (5 points)

- Automatic recalculation on review submit
- Update ServiceProviderProfile.AverageRating

- Update ServiceProviderProfile.TotalReviews

☐ **Admin Review Moderation (5 points)**

GET /api/admin/reviews

PUT /api/admin/reviews/{id}/hide

- Get all reviews
- Hide inappropriate reviews
- Review flag system

Frontend Tasks:

☐ **Review Submission Form (8 points)**

- Trigger after service completion
- Star rating input (1-5)
- Comment textarea
- Character counter (max 1000)
- Anonymous option checkbox
- Submit button
- Success confirmation

☐ **Provider Reviews Section (8 points)**

- Display on provider profile
- Average rating (large stars)
- Total reviews count
- Rating distribution chart
- Reviews list (paginated)
- Load more button

☐ **Review Card Component (3 points)**

- Reviewer name/avatar
- Star rating
- Review text
- Date posted
- "Helpful" button (future)

☐ **User Reviews Page (5 points)**

- List user's submitted reviews
- Edit review (within 7 days)
- View associated service request

☐ **Admin Review Moderation (5 points)**

- List all reviews
- Filter by rating, date
- Hide review button
- View associated request/provider

Testing:

☐ Backend tests (5 points)

☐ Frontend tests (3 points)

Deliverables:

- ☒ Review submission system
- ☒ Rating calculation
- ☒ Review display
- ☒ Admin moderation

Story Points: 60 **Team Capacity:** 160 hours

Sprint 9: Admin Dashboard & Reporting (Week 10)

Sprint Goals:

- Admin dashboard with metrics
- User management
- Provider management
- Booking management
- Reports and analytics

Backend Tasks:

Dashboard Metrics

☐ **Get Dashboard Stats** (8 points)

GET /api/admin/dashboard/stats

- Total users (active, inactive)
- Total providers (approved, pending, rejected)
- Total bookings (this month, all time)
- Total revenue (this month, all time)
- Average platform rating
- Pending applications count

☐ **Get Revenue Chart Data** (5 points)

GET /api/admin/reports/revenue

?period=month&year=2025

- Revenue by day/week/month
- Revenue by category

☐ **Get Booking Chart Data** (5 points)

GET /api/admin/reports/bookings

?period=month

- Bookings over time
- Booking status breakdown

User Management

☐ **Get All Users** (3 points)

GET /api/admin/users

?page=1&pageSize=20&search=

- Paginated user list
- Search by name, email

☐ **Get User Details** (2 points)

GET /api/admin/users/{id}

- Full user details
- Booking history

☐ **Suspend/Activate User** (3 points)

PUT /api/admin/users/{id}/suspend

PUT /api/admin/users/{id}/activate

- Toggle user status

Provider Management

☐ Get All Providers (3 points)

GET /api/admin/providers

?status=Approved&page=1

- Filter by status

- Search by name

☐ Suspend/Activate Provider (3 points)

PUT /api/admin/providers/{id}/suspend

- Suspend provider account

- Notify provider

Booking Management

☐ Get All Bookings (3 points)

GET /api/admin/requests

?status=&page=1

- All service requests

- Filter by status, date, provider

☐ Cancel Booking (3 points)

PUT /api/admin/requests/{id}/cancel

- Admin cancel booking

- Trigger refund if paid

Frontend Tasks:

☐ Admin Dashboard Page (21 points)

- Metrics Cards:

- Total users

- Total providers
- Total bookings
- Total revenue
- Average rating
- Pending applications
- **Charts:**
 - Revenue over time (line chart)
 - Bookings over time (area chart)
 - Revenue by category (bar chart)
 - Booking status (pie chart)
- **Recent Activity:**
 - Latest bookings
 - New applications
 - Recent reviews

☐ **User Management Page (8 points)**

- User list table
- Search and filters
- View user details modal
- Suspend/activate buttons
- Pagination

☐ **Provider Management Page (8 points)**

- Provider list table
- Filter by status
- View provider details
- Suspend/activate
- Performance metrics

☐ **Booking Management Page (8 points)**

- Bookings list table
- Status filters
- View booking details

- Cancel booking

- Refund button

☐ **Reports Page (8 points)**

- Date range selector
- Export to CSV/Excel
- Revenue report
- Provider performance report
- User activity report

Testing:

☐ Backend tests (5 points)

☐ Frontend tests (5 points)

Deliverables:

- ☒ Admin dashboard
- ☒ User management
- ☒ Provider management
- ☒ Booking management
- ☒ Reports and analytics

Story Points: 102 **Team Capacity:** 160 hours

Sprint 10: Testing, Bug Fixes & Deployment (Week 11)

Sprint Goals:

- Comprehensive testing (unit, integration, E2E)
- Bug fixing
- Performance optimization
- Production deployment
- Documentation

Tasks:

