# RepareerSimpel Platform Technical Plan

06/07/2025

## 1. Project Overview

- **Project Goal**: Launch functional repair booking platform by October 2025
- **Key Success Metrics**: full booking flow, voucher system operational
- **Target Timeline**: 12 weeks to launch

## 2. Technical Architecture Overview

**System Architecture Diagram**

- Frontend (web application)
- Backend API services
- Database layer
- Payment processing integration
- Email notification system

**Technology Stack Decisions**

- **Frontend**: React/Next.js ( fast development, SEO-friendly)
- **Backend**: Python FastAPI (rapid API development)
- **Database**: PostgreSQL (relational data, GDPR compliance)
- **Payments**: Stripe/Mollie (Dutch market, iDeal support)
- **Infrastructure**: Cloud hosting

## 3. Feature Development Roadmap

**Phase 1: Core Platform Foundation (Weeks 1-4)**

- **Week 1-2**: Project setup, database design, basic API structure

- **Week 3-4**: User authentication, basic shop management, search functionality

**Phase 2: Booking & Payment System (Weeks 5-8)**

- **Week 5-6**: Booking system, calendar integration, time slot management
- **Week 7-8**: Payment processing, voucher system, email notifications

**Phase 3: User Experience & Polish (Weeks 9-12)**

- **Week 9**: Frontend development incl. map integration, reviews
- **Week 10**: admin dashboard, shop onboarding tools, test deployment
- **Week 11**: testing & bug fixes
- **Week 12**: User acceptance testing, final bug fixes, deployment

**Phase 4: Post-Launch Monitoring (Week 13+)**

- **Week 13+**: continuous monitoring, updates & fixes

## 4. Technical Implementation Details

**Abridged Database Schema Design**

- Users table (consumers, shop owners, admins)
- Shops table (location, services, availability)
- Services table: device type, problem type, description.
- Bookings table (appointments, status tracking)
- Vouchers table (codes, usage tracking, municipality budgets)
- Reviews table (ratings, comments)

**API Endpoints Architecture**

- /api/search - Device/problem/location-based shop discovery
- /api/booking - Appointment creation and management

- /api/payment - Payment processing with voucher discounts
- /api/shop - Shop management and onboarding
- /api/admin - Administrative functions

**Security & Compliance**

- **GDPR Compliance**: Data encryption, user consent, deletion rights
- **Payment Security**: PCI DSS compliance, tokenization
- **Data Protection**: SSL/TLS, secure authentication, input validation

## 5. Integration Requirements

**Payment Processing**

- iDeal integration for Dutch users
- Optional credit card processing for international users
- Voucher discount calculation and tracking
- Municipality reimbursement tracking

**Third-Party Services**

- Google Maps API for location services
- Email service provider (Postmark/SendGrid/Mailgun)
- Analytics integration (Google Analytics/Pixel)
- Error tracking and alerting (Sentry)

**Municipality Integration**

- Voucher distribution mechanism
- Usage reporting and analytics

## 6. Testing Strategy

**Automated Testing**

- Unit tests for core business logic

- Integration tests for API endpoints
- End-to-end tests for critical user flows

**Manual Testing**

- Payment flow testing with test vouchers
- Cross-browser and mobile compatibility testing
- Ideally user acceptance testing with real shops

**Performance Testing**

- Load testing for concurrent bookings
- Database performance optimization
- API response time monitoring

## 7. Deployment & Infrastructure

**Hosting Strategy**

- Cloud-based infrastructure for scalability
- Staging environment for testing
- Production environment with monitoring

**CI/CD Pipeline**

- Automated testing on code commits
- Staging deployment for review
- Production deployment with rollback capability

**Monitoring & Maintenance**

- Application performance monitoring
- Error tracking and alerting
- Database backup and recovery procedures

## 8. Resource Requirements

**External Dependencies**

- Payment provider setup and testing
- Google Maps API access
- Email service provider setup
- Hosting environment configuration
- Sentry performance and error monitoring

## 9. Success Metrics & Acceptance Criteria

**Technical Metrics**

- 99% uptime during launch period
- <2 second page load times
- Zero critical security vulnerabilities
- 100% test coverage for payment flows

## 10. Post-Launch Support Plan

**Immediate Support (First 30 Days)**

- 24/7 monitoring
- Rapid bug fix deployment
- Weekly performance reviews

**Ongoing Maintenance**

- To be discussed.

# Technical requirements draft

## Frontend

- Municipality (Amsterdam) branding visibly on the landing page.
- Landing page: device type, problem type, and optional location input.
- Search results page: show repair shops in a list and on a map.
- Search results; shop card: display shop name, photo, certifications, star rating, price, repair duration, and opening hours.
- Search results: allow sorting by price, duration, rating.
- Search results: show selected search query and allow editing.
- Search results: "Book now" button for direct navigation to booking page.
- Shop page: shop description, services, photo, reviews.
- Booking page: show selected repair, shop name, and price.
- Booking page: user inputs name, surname, email, and optional phone.
- Booking page: select/confirm date and timeslot.
- Booking confirmation page and thank you message after payment success.
- Confirmation email to consumer after booking.
- Email to repair shop after booking.
- Reminder email (morning of repair day? day before?).
- (?) Google/Apple calendar invite attached to confirmation/reminder.
- Review request email after repair.
- 404 page with basic navigation.
- Privacy policy page.
- Terms of service page.
- About us page.
- Unified webpage shell with header/footer with relevant links.

- (?) Dual language support (Dutch and English).
- (?) Support/contact page with email and phone number.
- (?) FAQ page.

## Backend / API

- API to fetch shops based on device/problem and optional location.
- API to fetch and manage available booking slots per shop.
- API to submit and confirm bookings, including confirmation email.
- API to apply voucher discount and calculate total price.
- API to process payment via Stripe/Mollie.
- API to create and update repair shop data: shop description, services, prices, and opening hours.
- (?) API to manage voucher status and usage tracking.
- API to fetch reviews for a store.
- API to fetch available devices, problems, and categories for a store.
- API to generate post-repair review link/token.

## Database

- Users table: name, email, optional phone, type.
- Shops table: name, description, contact, location, photo, opening hours.
- Services table: device type, problem type, description.
- Shops_services table: shop, service, duration, price.
- Bookings table: user, shop, service, timeslot, status.
- Reviews table: shop, user, stars, comment, timestamp.
- Vouchers table: code, discount amount, redeemed, user link, timestamp.
- Diagnostics table: shop, consumer, quote sent, quote approved.
- Admin table: platform staff with privileges.
- (?) Log table for fraud analysis: IPs, MACs, voucher use.

## Admin Dashboard

- Add/edit/delete/set inactive repair shops.

- Toggle voucher campaigns on/off.
- Edit shop details: photos, services, prices, description.
- View booking stats and voucher usage per shop.
- Export CSV of shop bookings.
- (?) View fraud log data (IP reuse, voucher reuse).
- (?) Upload analytics scripts or pixels (e.g. Meta Pixel).

## Payments & Vouchers

- Payment processing via Stripe/Mollie with iDeal and credit card.
- Apply discount voucher during payment.
- Voucher validation (one per user, one per payment method).
- Refund logic for diagnostic flow (send quote email + payment link).
- Track total voucher budget (e.g. [5.1, 2, b] cap for 750 vouchers).
- Track redemption count per voucher.
- Payment success/fail callback handling.
- Auto-generate invoice per booking.

## DevOps & Deployment

- CI/CD pipeline with staging and production environments.
- Automated tests on commit (unit + integration).
- Rollback support for production deploys.
- Cloud-based hosting with autoscaling.
- Staging environment URL for manual review.
- Production environment with uptime monitoring.
- Nightly backups of PostgreSQL database.

## Security & Compliance

- Enforce HTTPS across the platform.
- Input validation on all form fields.
- GDPR-compliant data storage (user deletion on request).
- API rate limiting and abuse prevention.
- TLS for API communication.

- Cookie consent banner.
- Store logs of voucher use with user-identifiable metadata (IP, email).

## Fraud Prevention

- Log all voucher redemptions with timestamp and device metadata.
- Only one voucher per IP, email, name, phone, bank account.
- (?) Flag and log suspicious voucher use patterns.
- (?) Flag bookings from same IP/email within short time window.
- (?) Admin view to review flagged voucher transactions.
- (?) Initiate random audit calls to shops/customers.

## Analytics & Tracking

- Integrate Google Analytics or Plausible.
- Log booking flow drop-off per step.
- Log completed bookings per day.
- Log voucher usage over time.
- Admin dashboard showing key KPIs: bookings/day, avg discount, avg repair time.