

Roam Homeowner Property Dashboard

Product Requirements Document

1. Executive Summary

Vision

Create an addictive, Zillow-meets-Robinhood experience for homeowners with assumable mortgages. The dashboard transforms a static asset (their home) into a dynamic financial instrument they want to monitor regularly, showing them exactly how valuable their low mortgage rate is to potential buyers.

Core Value Proposition

"Your 2.75% mortgage rate could save a buyer \$847/month. See how much your rate is worth."

Growth Hypothesis

Homeowners who understand the dollar value of their rate advantage are 3-5x more likely to list with Roam. Daily/monthly engagement creates mental availability and positions Roam as the obvious choice when they decide to sell.

2. User Stories

Primary User: Homeowner with FHA/VA Assumable Mortgage

As a homeowner with a low-rate assumable mortgage,
I want to understand exactly how valuable my rate is to buyers,
So that I can make an informed decision about when and how to sell.

As a homeowner considering selling,
I want to see real-time demand from buyers seeking my rate,
So that I feel confident there's a market for my home.

As a curious homeowner,
I want to compare my rate to my neighbors and the market,
So that I understand my competitive position.

As an engaged homeowner,
I want regular updates on my home's value and rate advantage,
So that I can track my wealth building over time.

3. Feature Specification

3.1 Hero Section: The Rate Advantage Card

Purpose: Immediately show the homeowner the dollar value of their rate advantage.

Display Elements:

typescript

```
interface RateAdvantageCard {  
    currentMortgageRate: number;      // e.g., 2.75%  
    currentMarketRate: number;        // e.g., 6.85%  
    rateDifferential: number;        // e.g., 4.10%  
    monthlyPaymentSavings: number;    // e.g., $847/month  
    lifetimeSavings: number;         // Over remaining loan term  
    buyerPurchasingPowerBoost: number; // How much more house buyer can afford  
    rateAdvantageScore: number;       // 1-100 score for gamification  
}
```

Visual Design:

- Large, bold typography showing savings
- Animated counter that updates with market rate changes
- Color gradient from green (great rate) to neutral
- "Your rate is better than X% of homeowners" social proof

Calculations:

typescript

```

// Monthly savings calculation
const monthlyPaymentAtMarketRate = calculatePayment(
  remainingBalance,
  currentMarketRate,
  remainingTermMonths
);
const currentMonthlyPayment = calculatePayment(
  remainingBalance,
  currentMortgageRate,
  remainingTermMonths
);
const monthlySavings = monthlyPaymentAtMarketRate - currentMonthlyPayment;

// Purchasing power boost
// Amount of additional home price buyer can afford with lower rate
const purchasingPowerBoost = calculateMaxPurchasePrice(
  buyerMonthlyBudget,
  currentMortgageRate
) - calculateMaxPurchasePrice(
  buyerMonthlyBudget,
  currentMarketRate
);

```

3.2 Home Value & Equity Tracker

Purpose: Show wealth accumulation over time, creating emotional investment.

Display Elements:

typescript

```

interface EquityTracker {
    currentHomeValue: number;           // Estimated market value
    originalPurchasePrice: number;
    appreciation: number;              // Dollar and percentage
    currentLoanBalance: number;
    totalEquity: number;                // Value - Balance
    equityPercentage: number;           // Equity / Value
    monthlyEquityGain: number;          // Principal paydown + appreciation
    equityHistory: EquityDataPoint[];   // For charting
}

```

```

interface EquityDataPoint {
    date: Date;
    homeValue: number;
    loanBalance: number;
    equity: number;
}

```

Visualizations:

- Area chart showing equity growth over time
- Stacked visualization: principal paydown vs appreciation
- "You've built \$X in equity this month" callout
- Comparison to renting scenario ("If you'd rented, you'd have \$0 equity")

3.3 Neighborhood Rate Comparison

Purpose: Create social proof and FOMO by showing how their rate compares to neighbors.

Display Elements:

typescript

```

interface NeighborhoodComparison {
  userRate: number;
  neighborhoodAverageRate: number;
  neighborhoodMedianRate: number;
  rateDistribution: RateBucket[];    // Histogram data
  percentilRank: number;           // "Better than X% of neighbors"
  neighborsWithAssumableLoans: number;
  recentSales: RecentSale[];
}

interface RateBucket {
  rateRange: string;            // e.g., "2.5-3.0%"
  count: number;
  isUserBucket: boolean;
}

interface RecentSale {
  address: string;             // Anonymized: "123 Main St"
  salePrice: number;
  saleDate: Date;
  wasAssumable: boolean;
  rateIfKnown?: number;
  premiumOverMarket?: number;   // If sold with Roam
}

```

Visualizations:

- Histogram of neighborhood rates with user's position highlighted
- Map view showing rate heat map (anonymized)
- "X homes near you have assumable mortgages"
- Recent Roam transactions in area with premiums achieved

3.4 Buyer Demand Signals

Purpose: Create urgency by showing real buyer interest in their rate bracket.

Display Elements:

typescript

```

interface BuyerDemandSignals {
  activeBuyersInMarket: number;      // Buyers on Roam seeking this area
  buyersSeekingThisRateRange: number; // Within 0.5% of user's rate
  averageDaysToContract: number;    // For similar listings
  demandTrend: 'increasing' | 'stable' | 'decreasing';
  demandScore: number;             // 1-100
  buyerTestimonials: BuyerQuote[];   // Anonymous quotes
  savedSearchesMatching: number;    // Buyers who would match this home
}


```

```

interface BuyerQuote {
  quote: string;                  // "I'd pay a premium for a 3% rate"
  buyerProfile: string;           // "First-time buyer in [City]"
  targetRate: number;
}


```

Visualizations:

- Animated pulse showing "live" buyer activity
 - "X buyers are looking for rates like yours right now"
 - Demand trend chart over past 30/60/90 days
 - Notification: "A buyer just saved a search matching your home"
-

3.5 Scenario Calculator

Purpose: Let homeowners model different outcomes to drive consideration.

Scenarios to Model:

```
typescript
```

```
interface ScenarioCalculator {  
    // Scenario 1: Sell with Roam (Assumable)  
    sellWithRoam: {  
        estimatedSalePrice: number;  
        ratePremium: number;          // Premium from rate advantage  
        netProceeds: number;  
        roamFees: number;  
        timeToClose: string;  
        buyerPoolSize: number;  
    };  
  
    // Scenario 2: Traditional Sale  
    traditionalSale: {  
        estimatedSalePrice: number;  
        netProceeds: number;  
        agentFees: number;  
        timeToClose: string;  
    };  
  
    // Scenario 3: Keep & Refinance (Don't do it!)  
    refinanceScenario: {  
        newPayment: number;  
        paymentIncrease: number;  
        lifetimeCostIncrease: number;  
        warningMessage: string;      // "You'd lose $X in rate value"  
    };  
  
    // Scenario 4: Rent It Out  
    rentalScenario: {  
        estimatedRent: number;  
        monthlyCashFlow: number;  
        capRate: number;  
        keepRateAdvantage: boolean;  
    };  
  
    // Scenario 5: Seller Financing (Future Roam Product)  
    sellerFinancingScenario: {  
        monthlyIncome: number;  
        totalInterestEarned: number;  
        riskAssessment: string;  
    };  
}
```

Interactive Elements:

- Sliders for sale price adjustments
 - Toggle between scenarios
 - Side-by-side comparison view
 - "Your rate is worth \$X - don't lose it" messaging for refinance scenario
-

3.6 Activity Feed & Notifications

Purpose: Drive regular engagement through timely, relevant updates.

Feed Items:

```
typescript
```

```

type ActivityFeedItem =
| MarketRateUpdate
| HomeValueUpdate
| BuyerActivityAlert
| NeighborhoodSaleAlert
| EquityMilestone
| RateAnniversary
| SeasonalInsight;

interface MarketRateUpdate {
  type: 'market_rate_update';
  previousRate: number;
  newRate: number;
  impactOnSavings: number;           // "Your rate now saves buyers $X more"
  timestamp: Date;
}

interface HomeValueUpdate {
  type: 'home_value_update';
  previousValue: number;
  newValue: number;
  changePercent: number;
  newEquity: number;
  timestamp: Date;
}

interface BuyerActivityAlert {
  type: 'buyer_activity';
  message: string;                  // "5 new buyers searching your area"
  buyerCount: number;
  timestamp: Date;
}

interface EquityMilestone {
  type: 'equity_milestone';
  milestone: string;                // "$100K equity reached!"
  currentEquity: number;
  timestamp: Date;
}

```

Notification Strategy:

typescript

```

interface NotificationConfig {
    // High-value triggers (push + email)
    highPriority: [
        'significant_rate_movement',      // Market rates move 0.25%+
        'buyer_inquiry',                // Direct interest in their home
        'equity_milestone',             // $50K, $100K, etc.
        'neighbor_sold_with_roam',      // Social proof
    ];
    // Medium-value (email digest)
    mediumPriority: [
        'weekly_market_update',
        'monthly_equity_report',
        'new_buyers_in_area',
    ];
    // Low-value (in-app only)
    lowPriority: [
        'minor_value_change',
        'general_market_news',
    ];
}

```

3.7 Sharing & Viral Mechanics

Purpose: Turn homeowners into Roam evangelists.

Shareable Elements:

typescript

```

interface ShareableContent {
    // Rate Advantage Card (Social sharing)
    rateAdvantageCard: {
        imageUrl: string;           // Generated OG image
        shareText: string;          // "My mortgage rate could save a buyer $847/mo 🏠"
        shareUrl: string;           // Links to Roam with referral code
    };
}

// Neighborhood Report
neighborhoodReport: {
    pdfUrl: string;              // "Share with your neighbors"
    emailTemplate: string;
    shareMessage: string;
};

// Referral Program
referral: {
    referralCode: string;
    referralUrl: string;
    rewardsEarned: number;
    referralsCompleted: number;
};
}

```

Viral Loops:

1. "Send this report to your neighbors" - PDF with anonymized neighborhood rates
2. "Share your rate advantage" - Social card for Instagram/Facebook
3. "Know someone with an FHA/VA loan?" - Referral program
4. "See if your home qualifies" - Shareable qualification checker

3.8 Gamification & Engagement

Purpose: Create habit-forming engagement patterns.

Gamification Elements:

typescript

```

interface GamificationSystem {
    // Rate Advantage Score (like credit score)
    rateAdvantageScore: {
        score: number;           // 1-100
        factors: ScoreFactor[];
        trend: 'improving' | 'stable' | 'declining';
        percentileRank: number;
    };

    // Streaks
    engagementStreak: {
        currentStreak: number;      // Days checked in
        longestStreak: number;
        streakRewards: Reward[];
    };

    // Achievements
    achievements: Achievement[];

    // Leaderboard (optional, anonymized)
    neighborhoodRanking: number;    // "Top 10% best rates in your ZIP"
}

interface Achievement {
    id: string;
    name: string;          // "Rate Rockstar"
    description: string;    // "Your rate is in the top 5%"
    icon: string;
    earnedDate?: Date;
    progress?: number;       // 0-100 for incomplete
}

```

4. Data Architecture

4.1 Data Sources

typescript

```
interface DataSources {  
    // Property Data  
    propertyData: {  
        source: 'ATTOM' | 'CoreLogic' | 'Zillow' | 'Redfin';  
        dataPoints: [  
            'current_value_estimate',  
            'historical_values',  
            'property_characteristics',  
            'tax_assessments',  
            'comparable_sales',  
        ];  
        refreshFrequency: 'daily';  
    };  
  
    // Mortgage Data  
    mortgageData: {  
        source: 'User Input' | 'Mortgage Statement Upload' | 'Plaid';  
        dataPoints: [  
            'original_loan_amount',  
            'current_balance',  
            'interest_rate',  
            'loan_type', // FHA, VA, Conventional  
            'origination_date',  
            'monthly_payment',  
        ];  
        refreshFrequency: 'monthly' | 'on_statement_upload';  
    };  
  
    // Market Rate Data  
    marketRates: {  
        source: 'Freddie Mac' | 'Mortgage News Daily' | 'Optimal Blue';  
        dataPoints: [  
            'current_30yr_fixed',  
            'current_fha_rate',  
            'current_va_rate',  
            'historical_rates',  
        ];  
        refreshFrequency: 'daily';  
    };  
  
    // Roam Internal Data  
    roamData: {  
        dataPoints: [  
    
```

```
'active_buyers_by_market',
'buyer_rate_preferences',
'recent_transactions',
'average_premiums_achieved',
];
refreshFrequency: 'real-time';
});
}
```

4.2 Data Models

typescript

```
// Core Property Model

interface Property {
    id: string;
    address: Address;
    characteristics: PropertyCharacteristics;
    valuation: PropertyValuation;
    mortgage: MortgageDetails;
    owner: Owner;
    rateAdvantage: RateAdvantageMetrics;
    engagement: EngagementMetrics;
    createdAt: Date;
    updatedAt: Date;
}

interface Address {
    street: string;
    unit?: string;
    city: string;
    state: string;
    zip: string;
    county: string;
    latitude: number;
    longitude: number;
}

interface PropertyCharacteristics {
    bedrooms: number;
    bathrooms: number;
    squareFeet: number;
    lotSize: number;
    yearBuilt: number;
    propertyType: 'single_family' | 'condo' | 'townhouse' | 'multi_family';
    features: string[];
}

interface PropertyValuation {
    currentEstimate: number;
    estimateDate: Date;
    confidenceScore: number;          // Low/Medium/High
    priceHistory: PricePoint[];
    comparables: Comparable[];
}
```

```
interface MortgageDetails {  
    loanType: 'FHA' | 'VA' | 'Conventional' | 'USDA';  
    isAssumable: boolean;  
    originalAmount: number;  
    currentBalance: number;  
    interestRate: number;  
    originationDate: Date;  
    maturityDate: Date;  
    monthlyPayment: number;  
    escrowAmount: number;  
    lender: string;  
    loanNumber?: string;           // For verification  
}
```

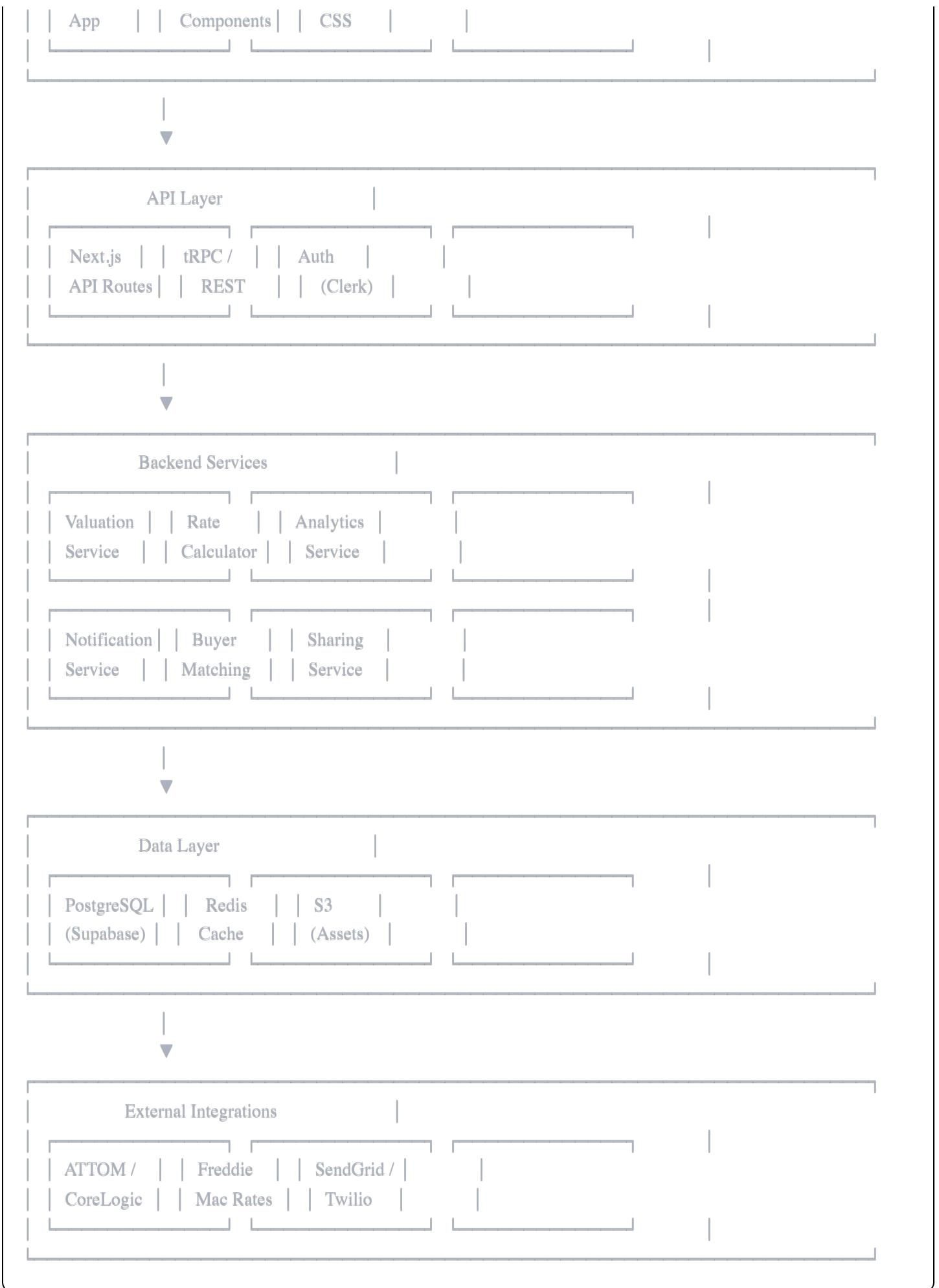
```
interface RateAdvantageMetrics {  
    currentMarketRate: number;  
    rateDifferential: number;  
    monthlyPaymentSavings: number;  
    lifetimeSavings: number;  
    purchasingPowerBoost: number;  
    rateAdvantageScore: number;  
    neighborhoodPercentile: number;  
    lastCalculated: Date;  
}
```

```
interface EngagementMetrics {  
    lastVisit: Date;  
    visitCount: number;  
    currentStreak: number;  
    scenariosRun: number;  
    sharesGenerated: number;  
    referralsSent: number;  
    notificationPreferences: NotificationPreferences;  
}
```

5. Technical Architecture

5.1 System Overview





5.2 Tech Stack Recommendation

typescript

```
const techStack = {
  frontend: {
    framework: 'Next.js 14 (App Router)',
    ui: 'React 18 + Tailwind CSS',
    components: 'sharden/ui',
    charts: 'Recharts or Tremor',
    animations: 'Framer Motion',
    state: 'Zustand or Jotai',
    forms: 'React Hook Form + Zod',
  },
  backend: {
    runtime: 'Node.js',
    api: 'Next.js API Routes or tRPC',
    database: 'PostgreSQL (Supabase or Neon)',
    cache: 'Redis (Upstash)',
    queue: 'Inngest or Trigger.dev',
    auth: 'Clerk or NextAuth',
  },
  infrastructure: {
    hosting: 'Vercel',
    cdn: 'Vercel Edge',
    storage: 'S3 / Cloudflare R2',
    monitoring: 'Vercel Analytics + Sentry',
    email: 'SendGrid or Resend',
    sms: 'Twilio',
  },
  dataProviders: {
    propertyData: 'ATTOM or CoreLogic API',
    mortgageRates: 'Freddie Mac PMMS or Mortgage News Daily',
    homeValues: 'Zillow API or HouseCanary',
  },
};
```

6. UI/UX Specifications

6.1 Page Structure

HEADER

HERO: RATE ADVANTAGE CARD

Your Rate: 2.75% Market Rate: 6.85%

Your rate saves buyers \$847/month

[Rate Advantage Score: 94/100]

Better than 97% of homeowners

[Share Your Rate Advantage] [See What Buyers Pay]

HOME VALUE & EQUITY

BUYER DEMAND

Home Value: \$485,000

▲ \$52,000 (+12%)

23 buyers searching

for your rate range

Your Equity: \$142,000

Avg days to contract: 18

[Equity Growth Chart]

[View Buyer Activity]

NEIGHBORHOOD COMPARISON

[Rate Distribution Histogram]

Your ZIP: 78704

Average Rate: 4.82% Your Rate: 2.75% ✓

Homes with Assumable Loans: 847

SCENARIO CALCULATOR

[Sell with Roam] [Traditional Sale] [Keep & Rent] [Refi]

Sell with Roam:

Estimated Sale Price: \$485,000

Rate Premium: + \$15,000

Roam Fee: - \$5,000

Net to You: \$152,000

[Get Your Personalized Estimate]

ACTIVITY FEED

Today

• Market rates up 0.12% - your savings now \$867/mo

• 3 new buyers saved searches matching your home

Yesterday

• Home on Oak St sold with Roam - \$12K premium

This Week

• Your home value increased \$2,400

CTA FOOTER

Ready to unlock your rate's value?

[Talk to a Roam Expert]

[List My Home]

6.2 Mobile-First Design

typescript

```
const responsiveBreakpoints = {
  mobile: '< 640px', // Single column, stacked cards
  tablet: '640-1024px', // Two column grid
  desktop: '> 1024px', // Full layout as shown above
};
```

```
const mobileConsiderations = [
  'Rate Advantage Card is full-width hero',
  'Bottom navigation bar for key actions',
  'Pull-to-refresh for latest data',
  'Swipeable cards for different sections',
  'Push notifications for engagement',
  'Quick share button always visible',
];
```

6.3 Design System

typescript

```

const designSystem = {
  colors: {
    primary: '#1E40AF',      // Roam Blue
    secondary: '#10B981',    // Success Green (for rate advantage)
    accent: '#F59E0B',       // Amber (for CTAs)
    danger: '#EF4444',       // Red (for warnings like refinance)
    neutral: {
      50: '#F9FAFB',
      100: '#F3F4F6',
      // ...full scale
      900: '#111827',
    },
  },
  typography: {
    hero: 'text-4xl font-bold',      // Rate savings number
    heading: 'text-2xl font-semibold', // Section headers
    subheading: 'text-lg font-medium', // Card titles
    body: 'text-base',               // Regular text
    caption: 'text-sm text-gray-500', // Secondary info
  },
  spacing: {
    section: 'py-8 px-4 md:px-6 lg:px-8',
    card: 'p-6 rounded-xl shadow-sm border',
    stack: 'space-y-4',
  },
  animations: {
    countUp: 'Animate numbers counting up on load',
    pulse: 'Subtle pulse on live data indicators',
    slideIn: 'Cards slide in on scroll',
    celebration: 'Confetti for milestones',
  },
};

```

7. Component Specifications

7.1 Core Components

typescript

```
// components/rate-advantage-card.tsx
interface RateAdvantageCardProps {
  userRate: number;
  marketRate: number;
  monthlySavings: number;
  lifetimeSavings: number;
  score: number;
  percentile: number;
  onShare: () => void;
}
```

```
// components/equity-tracker.tsx
interface EquityTrackerProps {
  currentValue: number;
  purchasePrice: number;
  currentBalance: number;
  equityHistory: EquityDataPoint[];
  timeRange: '1M' | '6M' | '1Y' | 'ALL';
  onTimeRangeChange: (range: string) => void;
}
```

```
// components/neighborhood-comparison.tsx
interface NeighborhoodComparisonProps {
  userRate: number;
  rateDistribution: RateBucket[];
  averageRate: number;
  medianRate: number;
  assumableCount: number;
  recentSales: RecentSale[];
}
```

```
// components/buyer-demand.tsx
interface BuyerDemandProps {
  activeBuyers: number;
  buyersForRateRange: number;
  avgDaysToContract: number;
  trend: 'up' | 'stable' | 'down';
  demandScore: number;
}
```

```
// components/scenario-calculator.tsx
interface ScenarioCalculatorProps {
  property: Property;
```

```

scenarios: {
  roam: SellWithRoamScenario;
  traditional: TraditionalSaleScenario;
  rental: RentalScenario;
  refinance: RefinanceScenario;
};

onScenarioSelect: (scenario: string) => void;
onGetEstimate: () => void;
}

// components/activity-feed.tsx
interface ActivityFeedProps {
  items: ActivityFeedItem[];
  isLoading: boolean;
  onLoadMore: () => void;
}

// components/share-modal.tsx
interface ShareModalProps {
  isOpen: boolean;
  onClose: () => void;
  content: ShareableContent;
  onShare: (platform: string) => void;
}

```

8. API Endpoints

8.1 REST API Structure

typescript

```

// Property Endpoints
GET /api/properties/:id           // Get property details
PUT /api/properties/:id           // Update property info
GET /api/properties/:id/valuation // Get latest valuation
GET /api/properties/:id/rate-advantage // Get rate advantage metrics
GET /api/properties/:id/equity-history // Get equity over time
GET /api/properties/:id/scenarios   // Get all scenario calculations

// Neighborhood Endpoints
GET /api/neighborhoods/:zipCode/rates // Get rate distribution
GET /api/neighborhoods/:zipCode/sales  // Get recent sales
GET /api/neighborhoods/:zipCode/stats  // Get market stats

// Buyer Demand Endpoints
GET /api/demand/:zipCode          // Get buyer demand signals
GET /api/demand/:zipCode/matching-buyers // Buyers matching property

// User Endpoints
GET /api/users/me                  // Get current user
PUT /api/users/me/notifications     // Update notification prefs
GET /api/users/me/activity         // Get activity feed
POST /api/users/me/referrals       // Generate referral link

// Sharing Endpoints
POST /api/share/rate-card          // Generate shareable card
POST /api/share/neighborhood-report // Generate PDF report

// Engagement Endpoints
POST /api/engagement/visit         // Log visit
POST /api/engagement/action        // Log action
GET /api/engagement/streak         // Get streak info
GET /api/engagement/achievements   // Get achievements

```

8.2 Webhook Handlers

typescript

```
// Triggered by external data updates
POST /api/webhooks/valuation-update      // Property value changed
POST /api/webhooks/rate-update           // Market rates changed
POST /api/webhooks/buyer-activity        // New buyer activity

// Triggered by Roam internal events
POST /api/webhooks/transaction-closed   // Nearby Roam transaction
POST /api/webhooks/buyer-saved-search    // Buyer saved matching search
```

9. Onboarding Flow

9.1 User Acquisition Funnel



STEP 3: Instant Results (The "Aha!" Moment)

Great news!

Your 2.75% rate could save a buyer

\$847/month

That's \$305,280 over the life of the loan!

[Create Free Account to Track This]

STEP 4: Account Creation

"Save your dashboard and get updates"

[Continue with Google]

[Continue with Apple]

or

Email: [_____]

Password: [_____]

Send me weekly rate advantage updates

Alert me when buyers search for my rate

[Create Account]

9.2 Progressive Data Collection

typescript

```
// Start with minimal data, progressively enhance
const dataCollectionStages = {
  stage1_anonymous: {
    required: ['address'],
    optional: [],
    unlocks: ['basic_valuation', 'neighborhood_rates'],
  },
  stage2_basic: {
    required: ['loan_type', 'interest_rate'],
    optional: ['loan_amount', 'origination_date'],
    unlocks: ['rate_advantage_score', 'monthly_savings'],
  },
  stage3_verified: {
    required: ['account_creation'],
    optional: ['mortgage_statement_upload', 'plaid_connection'],
    unlocks: ['full_dashboard', 'notifications', 'sharing'],
  },
  stage4_engaged: {
    required: [],
    optional: ['contact_info', 'timeline_to_sell'],
    unlocks: ['personalized_estimate', 'agent_connection'],
  },
};
```

10. Notification Strategy

10.1 Email Campaigns

typescript

```
const emailCampaigns = {  
  // Transactional  
  welcome: {  
    trigger: 'account_creation',  
    subject: 'Your rate advantage dashboard is ready',  
    content: 'Here's what your 2.75% rate is worth...',  
  },  
  
  // Engagement  
  weekly_digest: {  
    trigger: 'every_monday_9am',  
    subject: 'Your weekly rate advantage update',  
    content: 'Market rates moved, home value changed, buyer activity...',  
  },  
  
  monthly_report: {  
    trigger: 'first_of_month',  
    subject: 'Your {{month}} equity report',  
    content: 'You built $X in equity this month...',  
  },  
  
  // Re-engagement  
  dormant_7_days: {  
    trigger: 'no_visit_7_days',  
    subject: '3 buyers just searched for your rate range',  
    content: 'See what's happening with your rate advantage...',  
  },  
  
  dormant_30_days: {  
    trigger: 'no_visit_30_days',  
    subject: 'Your rate is now worth $X more',  
    content: 'Market rates increased, making your rate more valuable...',  
  },  
  
  // Event-triggered  
  rate_spike: {  
    trigger: 'market_rate_increase_0.25',  
    subject: '↗ Rates just jumped - your savings increased',  
    content: 'Your rate now saves buyers $X/month...',  
  },  
  
  neighbor_sold: {  
    trigger: 'roam_transaction_in_zip',  
  },
```

```
subject: 'Your neighbor just sold with Roam',  
content: 'A home on {{street}} captured a $X premium...',  
},  
  
equity_milestone: {  
  trigger: 'equity_crosses_threshold',  
  subject: '🎉 You hit $100K in equity!',  
  content: 'Celebrate your wealth building milestone...',  
},  
};
```

10.2 Push Notifications

typescript

```
const pushNotifications = {  
  high_priority: {  
    buyer_inquiry: {  
      title: 'A buyer is interested in your rate',  
      body: 'View their profile and respond',  
      action: 'open_buyer_detail',  
    },  
    significant_rate_move: {  
      title: 'Rates jumped 0.5% today',  
      body: 'Your savings increased to ${{amount}}/mo',  
      action: 'open_dashboard',  
    },  
  },  
  
  medium_priority: {  
    new_buyers: {  
      title: '{{count}} new buyers in your area',  
      body: 'They're looking for rates like yours',  
      action: 'open_buyer_demand',  
    },  
    value_update: {  
      title: 'Your home value updated',  
      body: '{{direction}} ${{amount}} this week',  
      action: 'open_equity_tracker',  
    },  
  },  
  
  engagement: {  
    streakReminder: {  
      title: "Don't lose your streak!",  
      body: '{{days}} days checking your rate advantage',  
      action: 'open_dashboard',  
    },  
    achievement_earned: {  
      title: '🏆 New achievement unlocked!',  
      body: '{{achievement_name}}',  
      action: 'open_achievements',  
    },  
  },  
};
```

11. Analytics & Success Metrics

11.1 Key Performance Indicators

typescript

```
const kpis = {  
  // Acquisition  
  signups_per_week: {  
    definition: 'New accounts created',  
    target: 'Week over week growth',  
  },  
  signup_conversion_rate: {  
    definition: 'Visitors who create accounts',  
    target: '> 15%',  
  },  
  
  // Engagement  
  dau_mau_ratio: {  
    definition: 'Daily active / Monthly active users',  
    target: '> 20% (strong engagement)',  
  },  
  weekly_active_rate: {  
    definition: 'Users who visit at least once per week',  
    target: '> 40%',  
  },  
  avg_session_duration: {  
    definition: 'Time spent per visit',  
    target: '> 3 minutes',  
  },  
  
  // Conversion  
  listing_conversion_rate: {  
    definition: 'Dashboard users who list with Roam',  
    target: '> 5%',  
  },  
  avg_time_to_list: {  
    definition: 'Days from signup to listing',  
    target: 'Track and optimize',  
  },  
  
  // Virality  
  referral_rate: {  
    definition: 'Users who send referrals',  
    target: '> 10%',  
  },  
  viral_coefficient: {  
    definition: 'New users from referrals / Total users',  
    target: '> 0.5',  
  },
```

```
},  
share_rate: {  
  definition: 'Users who share rate card',  
  target: '> 20%',  
},  
  
// Retention  
d7_retention: {  
  definition: 'Users returning after 7 days',  
  target: '> 30%',  
},  
d30_retention: {  
  definition: 'Users returning after 30 days',  
  target: '> 20%',  
},  
};
```

11.2 Event Tracking

typescript

```

const trackingEvents = {
  // Page Views
  'dashboard.view': { properties: ['source', 'time_since_last_visit'] },
  'scenario.view': { properties: ['scenario_type'] },
  'neighborhood.view': { properties: [] },

  // Interactions
  'scenario.calculate': { properties: ['scenario_type', 'inputs'] },
  'share.initiated': { properties: ['share_type', 'platform'] },
  'share.completed': { properties: ['share_type', 'platform'] },
  'cta.clicked': { properties: ['cta_type', 'location'] },

  // Engagement
  'notification.received': { properties: ['notification_type'] },
  'notification.clicked': { properties: ['notification_type'] },
  'email.opened': { properties: ['campaign'] },
  'email.clicked': { properties: ['campaign', 'link'] },

  // Conversion
  'listing.started': { properties: ['source'] },
  'listing.completed': { properties: ['source', 'time_to_convert'] },
  'consultation.booked': { properties: ['source'] },

  // Referral
  'referral.created': { properties: [] },
  'referral.clicked': { properties: ['referrer_id'] },
  'referral.converted': { properties: ['referrer_id'] },
};


```

12. MVP Scope vs Future Phases

12.1 MVP (Phase 1) - 4-6 weeks

Must Have:

- Address lookup and property identification
- Manual mortgage data entry
- Rate Advantage Card with savings calculation
- Basic home value display (single source)
- Simple equity calculation
- Account creation and authentication
- Basic email notifications (welcome, weekly digest)

- Mobile-responsive design
- Share rate card functionality

Technical MVP:

- Next.js app with basic pages
- Supabase for auth and database
- Single property data API integration
- Basic calculation engine
- SendGrid for emails

12.2 Phase 2 - Enhanced Engagement (4 weeks)

- Neighborhood rate comparison
- Rate distribution histogram
- Buyer demand signals (from Roam data)
- Scenario calculator (all 4 scenarios)
- Activity feed
- Push notifications
- Equity history chart
- Achievement system (basic)

12.3 Phase 3 - Viral Growth (4 weeks)

- Shareable neighborhood report PDF
- Referral program with tracking
- Social sharing optimization
- Rate anniversary celebrations
- Gamification (streaks, leaderboard)
- Mortgage statement upload (auto-fill)
- Enhanced notification personalization

12.4 Phase 4 - Advanced Features (6 weeks)

- Plaid integration for mortgage verification
- Multiple property support
- Seller financing scenario calculator
- Agent dashboard view
- API for partner integrations
- Advanced analytics and A/B testing
- Machine learning for engagement optimization

13. Security & Privacy

13.1 Data Protection

typescript

```
const securityMeasures = {
  authentication: {
    method: 'OAuth 2.0 + Magic Links',
    mfa: 'Optional SMS/Authenticator',
    session: 'JWT with refresh tokens',
  },
  dataEncryption: {
    atRest: 'AES-256',
    inTransit: 'TLS 1.3',
    sensitiveFields: ['loan_number', 'ssn_last_4'],
  },
  accessControl: {
    principle: 'Least privilege',
    userCanAccess: 'Only their own properties',
    adminAccess: 'Audit logged',
  },
  privacyCompliance: {
    regulations: ['CCPA', 'State privacy laws'],
    dataRetention: '7 years for transactions',
    rightToDelete: 'Supported',
    dataPortability: 'Export to JSON/CSV',
  },
};
```

13.2 Neighborhood Data Privacy

typescript

```

const neighborhoodPrivacy = {
  // What we show
  aggregatedData: [
    'rate_distribution_histogram',
    'average_rates_by_zip',
    'count_of_assumable_loans',
  ],
  // What we anonymize
  anonymizedData: [
    'recent_sales (address truncated)',
    'rate_percentiles (no individual rates)',
  ],
  // What we never show
  neverExpose: [
    'individual_neighbor_rates',
    'specific_addresses_with_rates',
    'personal_mortgage_details_of_others',
  ],
};

```

14. Implementation Notes for Cursor

14.1 Project Setup

```

bash

# Initialize Next.js project
npx create-next-app@latest roam-homeowner-dashboard --typescript --tailwind --eslint --app --src-dir

# Install core dependencies
npm install @supabase/supabase-js @clerk/nextjs recharts framer-motion
npm install react-hook-form @hookform/resolvers zod
npm install @tanstack/react-query axios date-fns
npm install -D @types/node

# Install UI components
npx shadcn-ui@latest init
npx shadcn-ui@latest add button card input label tabs dialog

```

14.2 Folder Structure

```
src/
  └── app/
    ├── (auth)/
    │   ├── login/
    │   └── signup/
    ├── (dashboard)/
    │   ├── layout.tsx
    │   ├── page.tsx      # Main dashboard
    │   ├── scenarios/
    │   └── neighborhood/
    │       └── settings/
    └── api/
        ├── properties/
        ├── valuations/
        ├── notifications/
        └── webhooks/
    └── onboarding/
  └── components/
    ├── dashboard/
    │   ├── rate-advantage-card.tsx
    │   ├── equity-tracker.tsx
    │   ├── buyer-demand.tsx
    │   ├── neighborhood-comparison.tsx
    │   ├── scenario-calculator.tsx
    │   └── activity-feed.tsx
    ├── ui/          # shadcn components
    └── shared/
  └── lib/
    ├── supabase/
    ├── calculations/
    │   ├── rate-advantage.ts
    │   ├── equity.ts
    │   └── scenarios.ts
    └── api/
        ├── property-data.ts
        └── market-rates.ts
    └── utils/
  └── hooks/
    ├── use-property.ts
    ├── use-rate-advantage.ts
    └── use-notifications.ts
  └── types/
```

```
|   └── property.ts  
|   └── mortgage.ts  
|   └── scenarios.ts  
└── styles/  
    └── globals.css
```

14.3 Key Implementation Priorities

typescript

```
const implementationOrder = [  
  // Week 1: Foundation  
  '1. Supabase schema and auth setup',  
  '2. Basic layout and routing',  
  '3. Address lookup component',  
  '4. Mortgage input form',  
  
  // Week 2: Core Calculations  
  '5. Rate advantage calculation engine',  
  '6. Rate Advantage Card component',  
  '7. Property data API integration',  
  '8. Basic home value display',  
  
  // Week 3: Dashboard  
  '9. Full dashboard layout',  
  '10. Equity tracker with chart',  
  '11. Scenario calculator (basic)',  
  '12. Mobile responsive design',  
  
  // Week 4: Engagement  
  '13. Account creation flow',  
  '14. Email notification setup',  
  '15. Share functionality',  
  '16. Activity feed (basic)',  
  
  // Week 5-6: Polish & Launch  
  '17. Onboarding flow optimization',  
  '18. Error handling and edge cases',  
  '19. Performance optimization',  
  '20. Analytics integration',  
];
```

15. Copy & Messaging Guidelines

15.1 Value Proposition Messaging

typescript

```
const messaging = {  
  headlines: {  
    primary: "Your mortgage rate is worth money. See how much.",  
    secondary: "Turn your low rate into a selling advantage.",  
    urgency: "Rates just hit 7%. Your 2.75% is gold.",  
  },  
  
  valueProps: {  
    savings: "Your rate saves buyers ${amount}/month",  
    premium: "Homes with rates like yours sell for ${amount} more",  
    demand: "${count} buyers are searching for your rate right now",  
    equity: "You've built ${amount} in equity",  
  },  
  
  ctas: {  
    primary: "See What Your Rate Is Worth",  
    secondary: "Get Your Free Estimate",  
    share: "Share Your Rate Advantage",  
    list: "List with Roam",  
  },  
  
  socialProof: {  
    transactions: "1,000+ families have sold with Roam",  
    savings: "Buyers have saved $50M+ through Roam",  
    premium: "Average seller premium: $15,000",  
  },  
};
```

15.2 Notification Copy

typescript

```
const notificationCopy = {  
  rate_increase: {  
    title: "↗️ Your rate just got more valuable",  
    body: "Market rates rose to {{rate}}%. Your {{userRate}}% now saves buyers ${{savings}}/month.",  
  },  
  
  buyer_activity: {  
    title: "🔥 {{count}} buyers searching your area",  
    body: "They're looking for rates below {{targetRate}}%. Yours qualifies.",  
  },  
  
  equity_milestone: {  
    title: "🎉 ${{amount}} equity milestone!",  
    body: "Your home is building wealth. See your full equity breakdown.",  
  },  
  
  neighbor_sold: {  
    title: "🏡 A neighbor just sold with Roam",  
    body: "They captured a ${{premium}} premium from their rate. Could you?",  
  },  
};
```

Appendix A: Calculation Formulas

typescript

```
// Monthly payment calculation (standard amortization)
function calculateMonthlyPayment(
    principal: number,
    annualRate: number,
    termMonths: number
): number {
    const monthlyRate = annualRate / 100 / 12;
    return principal *
        (monthlyRate * Math.pow(1 + monthlyRate, termMonths)) /
        (Math.pow(1 + monthlyRate, termMonths) - 1);
}

// Monthly savings from rate difference
function calculateMonthlySavings(
    balance: number,
    userRate: number,
    marketRate: number,
    remainingMonths: number
): number {
    const userPayment = calculateMonthlyPayment(balance, userRate, remainingMonths);
    const marketPayment = calculateMonthlyPayment(balance, marketRate, remainingMonths);
    return marketPayment - userPayment;
}

// Lifetime savings
function calculateLifetimeSavings(
    monthlySavings: number,
    remainingMonths: number
): number {
    return monthlySavings * remainingMonths;
}

// Purchasing power boost
function calculatePurchasingPowerBoost(
    targetMonthlyPayment: number,
    userRate: number,
    marketRate: number,
    termMonths: number
): number {
    const maxPriceAtUserRate = calculateMaxPurchasePrice(
        targetMonthlyPayment, userRate, termMonths
    );
    const maxPriceAtMarketRate = calculateMaxPurchasePrice(  
)
```

```

    targetMonthlyPayment, marketRate, termMonths
);
return maxPriceAtUserRate - maxPriceAtMarketRate;
}

// Rate advantage score (1-100)
function calculateRateAdvantageScore(
    userRate: number,
    marketRate: number,
    neighborhoodAvgRate: number
): number {
    const marketDiff = marketRate - userRate;
    const neighborDiff = neighborhoodAvgRate - userRate;

    // Weighted score based on market differential and neighborhood comparison
    const marketScore = Math.min(marketDiff * 15, 50); // Max 50 points
    const neighborScore = Math.min(neighborDiff * 20, 50); // Max 50 points

    return Math.round(Math.max(0, Math.min(100, marketScore + neighborScore)));
}

```

Appendix B: Database Schema

sql

-- Core tables

```
CREATE TABLE properties (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    user_id UUID REFERENCES users(id),
    address JSONB NOT NULL,
    characteristics JSONB,
    created_at TIMESTAMPTZ DEFAULT NOW(),
    updated_at TIMESTAMPTZ DEFAULT NOW()
);
```

```
CREATE TABLE mortgages (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    property_id UUID REFERENCES properties(id),
    loan_type VARCHAR(20) NOT NULL,
    is_assumable BOOLEAN DEFAULT false,
    original_amount DECIMAL(12,2),
    current_balance DECIMAL(12,2),
    interest_rate DECIMAL(5,3),
    origination_date DATE,
    maturity_date DATE,
    monthly_payment DECIMAL(10,2),
    verified BOOLEAN DEFAULT false,
    created_at TIMESTAMPTZ DEFAULT NOW(),
    updated_at TIMESTAMPTZ DEFAULT NOW()
);
```

```
CREATE TABLE valuations (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    property_id UUID REFERENCES properties(id),
    value_estimate DECIMAL(12,2),
    confidence_score VARCHAR(10),
    source VARCHAR(50),
    valuation_date DATE,
    created_at TIMESTAMPTZ DEFAULT NOW()
);
```

```
CREATE TABLE rate_snapshots (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    rate_type VARCHAR(20), -- '30yr_fixed', 'jha', 'va'
    rate DECIMAL(5,3),
    source VARCHAR(50),
    snapshot_date DATE,
    created_at TIMESTAMPTZ DEFAULT NOW()
```

```
);
```

```
CREATE TABLE activity_events (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    user_id UUID REFERENCES users(id),
    property_id UUID REFERENCES properties(id),
    event_type VARCHAR(50),
    event_data JSONB,
    created_at TIMESTAMPTZ DEFAULT NOW()
);
```

```
CREATE TABLE notifications (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    user_id UUID REFERENCES users(id),
    notification_type VARCHAR(50),
    title VARCHAR(200),
    body TEXT,
    data JSONB,
    sent_at TIMESTAMPTZ,
    read_at TIMESTAMPTZ,
    created_at TIMESTAMPTZ DEFAULT NOW()
);
```

```
CREATE TABLE user_engagement (
    id UUID PRIMARY KEY DEFAULT uuid_generate_v4(),
    user_id UUID REFERENCES users(id) UNIQUE,
    last_visit TIMESTAMPTZ,
    visit_count INTEGER DEFAULT 0,
    current_streak INTEGER DEFAULT 0,
    longest_streak INTEGER DEFAULT 0,
    scenarios_run INTEGER DEFAULT 0,
    shares_generated INTEGER DEFAULT 0,
    referrals_sent INTEGER DEFAULT 0,
    achievements JSONB DEFAULT '[]',
    created_at TIMESTAMPTZ DEFAULT NOW(),
    updated_at TIMESTAMPTZ DEFAULT NOW()
);
```

```
-- Indexes
```

```
CREATE INDEX idx_properties_user ON properties(user_id);
CREATE INDEX idx_mortgages_property ON mortgages(property_id);
CREATE INDEX idx_valuations_property ON valuations(property_id);
```

```
CREATE INDEX idx_activity_user ON activity_events(user_id);  
CREATE INDEX idx_notifications_user ON notifications(user_id);
```

Document Version: 1.0 Last Updated: December 2024 Author: Claude for Roam