Business Planner - Project Brief

🎯 Executive Summary

What: Voice-first task management system for an entrepreneur managing 4 business directions via Telegram bot.

Core Value: Transform voice messages into structured tasks with AI parsing, eliminating manual data entry while maintaining full control over task organization.

Key Innovation: Self-learning system that improves time estimates and task categorization based on historical data using RAG (Retrieval-Augmented Generation).

👤 User Context

The User

- **Name**: Константин
- **Role**: Founder/CEO managing all 4 business directions
- **Pain Points**:
  - Constant context switching between businesses
  - Many small tasks that are tedious to manually log
  - Need to delegate tasks to team members
  - Hard to track time spent across businesses
- **Workflow**: Primarily mobile, on-the-go, prefers voice input
- **Location**: Almaty, Kazakhstan (UTC+5)

The 4 Businesses

1. **Inventum** - Dental equipment repair service
   - Tasks: diagnostics, repairs, client visits, service calls
   - Team: Константин (CEO), Максим (директор), Дима (мастер), Максут (выездной мастер)
   - Keywords: ремонт, диагностика, выезд, сервис, оборудование
2. **Inventum Lab** - Dental laboratory
   - Tasks: modeling, milling, crown production, CAD/CAM
   - Team: Константин (CEO), Юрий Владимирович (директор), Мария (CAD/CAM оператор)
   - Keywords: моделирование, фрезеровка, коронки, протезы, CAD
3. **R&D** - Prototype development & research
   - Tasks: design, testing, documentation, prototyping
   - Location: Always "Workshop/Мастерская"
   - Team: Константин (CEO), Максим, Дима
   - Keywords: прототип, разработка, тестирование, исследование, датчик
4. **Import & Trade** - Equipment import from China
   - Tasks: supplier calls, logistics, customs, contracts
   - Team: Константин (CEO), Слава (юрист/бухгалтер)
   - Keywords: поставка, станок, Китай, таможня, договор, импорт

Cross-functional:

- **Константин** - CEO/Founder of all businesses
- **Лиза** - Marketing/SMM for all businesses
- Keywords for Лиза: маркетинг, реклама, SMM, пост, контент, продвижение

🔄 Core User Journey

Primary Flow: Voice to Task

```
1. USER SENDS VOICE MESSAGE
   "Завтра утром нужно позвонить поставщику фрез по проекту декабрьская поставка"

2. SYSTEM PROCESSES
   - Transcribes voice (Whisper API)
   - Extracts structured data
   - Determines business context (@trade)
   - Parses deadline ("завтра утром" → next workday 09:00)
   - Identifies project if mentioned
   - Estimates duration from similar past tasks

3. BOT RESPONDS
   "✅ Создал задачу:
   📞 Позвонить поставщику фрез
   📦 Бизнес: Импорт и торговля
   📁 Проект: Декабрьская поставка
   📅 Понедельник 09:00
   ⏱️ ~30 минут (на основе похожих звонков)"
```

Secondary Flows

Daily Planning

```
User: /today
Bot: Returns prioritized task list grouped by business
```

Project Creation

```
User: "Создай проект Ремонт фрезера Иванова для Inventum"
Bot: Creates project container for future tasks
```

Task Completion

```
User: "Выполнил задачу диагностика платы за 2 часа"
Bot: Marks complete, stores actual duration for future learning
```

📊 Data Model

Core Entities

python

```
Task:
  - title: str                    # What to do
  - business: str                 # One of 4 businesses (required)
  - project: str | None           # Optional project grouping
  - priority: 1-4                 # Based on importance × urgency
  - deadline: datetime            # Smart parsing with workday logic
  - estimated_duration: int       # AI estimate in minutes
  - actual_duration: int | None   # User feedback for learning
  - status: "open" | "done" | "archived"
  - embedding: vector[1536]       # For similarity search

Project:
  - name: str                     # User-defined name
  - business: str                 # Parent business
  - status: "active" | "on_hold" | "completed"

Business:
  - Fixed set of 4 (Inventum, Inventum Lab, R&D, Import & Trade)
  - Each has context-specific keywords for auto-detection

Member:
  - name: str                     # Константин, Максим, Дима, Максут, Юрий Владимирович, Мария, Слава, Лиза
  - businesses: List[str]         # Which businesses they work in
  - role: str                     # Their specialization
  - keywords: List[str]           # Trigger words for auto-assignment
```

🧠 AI Architecture

Three-Tier Model Strategy

Tier 1: GPT-4o-mini (Cheap, Fast)

- Task parsing from transcribed text
- Extract: business, deadline, project, participants
- Simple structured data extraction

Tier 2: GPT-4o (Balanced)

- Task categorization (energy level)
- Duration estimation
- Smart deadline interpretation
- Daily plan optimization

Tier 3: Claude 3.5 Sonnet (Premium, Rare)

- Weekly analytics and insights
- Strategic recommendations
- Complex pattern analysis
- Used sparingly (1-2 times per week)

RAG Pipeline

python

```python
# When creating new task:
1. Generate embedding for new task
2. Search similar past tasks (filtered by business!)
3. Adjust time estimate based on historical actual_duration
4. Store for future learning

# Critical: Business isolation
# "diagnostics" in @inventum ≠ "diagnostics" in @r&d
# Always filter vector search by business_id
```

🎯 Prioritization Logic

Simplified Eisenhower Matrix

Priority 1: DO NOW (Important + Urgent)
Priority 2: SCHEDULE (Important + Not Urgent)
Priority 3: DELEGATE (Not Important + Urgent)
Priority 4: BACKLOG (Not Important + Not Urgent)

Smart Defaults

- No deadline specified → +7 days
- No time specified → 23:59 same day
- Weekend deadline → Move to Monday
- "утром" → 09:00, "днем" → 13:00, "вечером" → 18:00

Auto-participant Assignment Rules

python

```
# Based on task keywords + business context
assignment_rules = {
    "@inventum": {
        "стратегия|план|бюджет": ["Константин"],
        "выезд|клиент|на место": ["Максут"],
        "ремонт|диагностика": ["Дима", "Максут"],
        "договор|документы": ["Максим"],
        "default": ["Дима"]
    },
    "@lab": {
        "стратегия|развитие": ["Константин"],
        "CAD|моделирование|дизайн": ["Мария"],
        "фрезеровка|производство": ["Мария"],
        "клиент|переговоры": ["Юрий Владимирович"],
        "default": ["Мария"]
    },
    "@r&d": {
        "концепция|стратегия": ["Константин"],
        "разработка|прототип": ["Максим", "Дима"],
        "тестирование": ["Дима"],
        "default": ["Максим", "Дима"]
    },
    "@trade": {
        "стратегия|партнеры": ["Константин"],
        "договор|юридический|таможня": ["Слава"],
        "поставщик|переговоры": ["Слава"],
        "бухгалтерия|счет": ["Слава"],
        "default": ["Слава"]
    },
    # Cross-functional
    "маркетинг|SMM|реклама|пост": ["Лиза"],
    # CEO tasks (если явно не указано "я")
    "я должен|мне нужно|я делаю": ["Константин"]
}
```

🔄 Self-Learning System

How It Learns

1. **Initial Estimate**: AI estimates 30 min for "call supplier"
2. **User Completion**: "Done in 45 minutes"
3. **System Learns**: Updates embedding with actual duration
4. **Next Time**: Similar task gets 45 min estimate
5. **Accuracy Improves**: System tracks estimate accuracy metric

Learning Boundaries

- Learning is isolated per business
- User-specific patterns (user's "вечером" = 19:00, not 18:00)
- Seasonal adjustments (December tasks take longer)

💡 Critical Business Rules

Task Creation

1. **Business Required**: Every task MUST have a business context
2. **Project Optional**: User explicitly mentions project or it's null
3. **Smart Participants**: Auto-assign based on task type + business
4. **Workday Respect**: Auto-adjust to working days (Mon-Fri)

Project Management

- Projects are NOT auto-created
- Projects are simple task groupings, nothing more
- User has full control over project creation
- No automatic decomposition

Context Isolation

- Each business is a separate context
- Same words can mean different things
- RAG search MUST filter by business
- Time estimates are business-specific

🚀 Implementation Priorities

Phase 1: MVP (Core Loop)

- □ Voice transcription → Task creation
- □ Basic deadline parsing
- □ Business auto-detection
- □ Simple /today command
- □ Task completion

Phase 2: Intelligence

- □ RAG similarity search
- □ Time estimation learning
- □ Project support
- □ Smart participant assignment
- □ Daily plan optimization

Phase 3: Analytics

- □ Weekly insights
- □ Time tracking across businesses

- □ Productivity patterns
- □ Strategic recommendations

🛠️ Technical Stack

Required

- **LangGraph**: Orchestration of AI pipeline
- **PostgreSQL + pgvector**: Database with vector search
- **Telegram Bot API**: User interface
- **OpenAI APIs**: Whisper (speech), GPT (parsing), Embeddings
- **FastAPI**: Backend API

Architecture Pattern

- Event-driven with LangGraph
- Each user action triggers a specific graph
- Stateful conversation with checkpoints
- Async processing throughout

📐 Success Metrics

1. **Task Creation Speed**: <10 seconds from voice to confirmation
2. **Parsing Accuracy**: >90% correct business/deadline extraction
3. **Time Estimate Accuracy**: Improve from 50% to 80% over 1 month
4. **User Effort**: 80% of tasks created without manual editing
5. **Daily Usage**: User creates 10+ tasks per day via voice

🎨 User Experience Principles

1. **Voice First**: Optimize for speech, not typing
2. **Forgiveness**: Easy to edit/correct AI mistakes
3. **Transparency**: Show why decisions were made
4. **Progressive**: Start simple, unlock complexity
5. **Contextual**: Respect business boundaries

🔑 Key Insights for Implementation

What Makes This Different

- **Not another todo app**: It's a voice-first AI assistant
- **Not project management**: It's task capture with smart grouping
- **Not time tracking**: It's predictive time estimation
- **Not automation**: User maintains full control

Critical Success Factors

1. Fast voice processing (<3 seconds)
2. Accurate business context detection
3. Smart but not clever (avoid over-engineering)

4. Learn from user, don't impose structure
5. Respect the hustle (entrepreneur mindset)

📝 Example Interactions

Morning Routine

```
User: "Что на сегодня?"
Bot: "Доброе утро! Сегодня 8 задач:

🔴 Срочные (до обеда):
- Диагностика платы для Иванова (@inventum) - 2ч
- Звонок поставщику (@trade) - 30м

🟡 На день:
- Моделирование коронок (@lab) - 1.5ч
..."
```

Personal Task

```
User: "Я должен подготовить презентацию для инвесторов по всем направлениям"
Bot: "✅ Подготовить презентацию для инвесторов
📦 Все направления | 👤 Константин | 📅 +7 дней | ⏱️ ~4 часа"
```

Quick Task

```
User: [voice] "Надо заказать фрезы из Китая до конца недели, пусть Слава оформит договор"
Bot: "✅ Заказать фрезы из Китая
📦 @trade | 👤 Слава | 📅 Пятница 18:00 | ⏱️ ~45 мин"
```

Project Task with Auto-assignment

User: "По проекту новый сайт нужно сделать моделирование коронок"
Bot: "✅ Сделать моделирование коронок
📦 @lab | 👤 Мария | 📁 Новый сайт | 📅 +7 дней | ⏱️ ~3 часа"

Cross-functional Task

📋

User: "Лиза должна сделать посты про новое оборудование для всех площадок"
Bot: "✅ Сделать посты про новое оборудование
📦 Все направления | 👤 Лиза | 📅 +7 дней | ⏱️ ~2 часа"

Service Call Task

📋

User: "Завтра выезд к клиенту на Абая ремонт фрезера"
Bot: "✅ Выезд ремонт фрезера
📦 @inventum | 👤 Максут | 📍 Абая | 📅 Завтра 09:00 | ⏱️ ~3 часа"

🎯 Final Note for Claude Code

This system is about **reducing friction** for a busy entrepreneur. Every decision should optimize for:

1. Speed of input (voice is fastest)
2. Minimal cognitive load (AI handles the parsing)
3. Practical intelligence (learn from usage, don't over-prescribe)
4. Business context respect (Inventum ≠ Lab ≠ R&D ≠ Trade)

The user should feel like they're talking to a smart assistant who knows their business, not filling out forms in a task management app.