Product Requirements Document: Scheduled Message Telegram Bot

1. Product Overview

A Telegram bot that sends scheduled messages to multiple private chats on a weekly basis. The bot operates autonomously without user interaction, sending predefined content according to configured schedules.

2. Core Functionality

2.1 Message Scheduling

Bot sends messages on a weekly schedule

Each chat has its own specific schedule (day of week + time)

All schedules are stored in UTC and converted to local chat timezones

Schedules persist after bot restarts

Configuration is managed through backend code/configuration files

2.2 Message Content

Primary content type: Text messages

Secondary content type: Images (stored locally)

Content is predefined and same for all chats

No message formatting required

2.3 Chat Management

Bot supports multiple private chats

No group chat support

Bot must be added to chats manually

No user interaction or commands supported

3. Technical Specifications

3.1 Storage

Schedule and chat configuration stored in JSON format

Structure:

```
{
    "chats": [
      {
        "chat_id": "123456789",
        "timezone": "Europe/London",
        "schedule": {
          "day_of_week": 1,
          "time": "10:00"
        }
      }
    ],
    "message_content": {
      "text": "Weekly message content",
      "image_path": "/path/to/image.jpg"  // Optional
    }
  }
```

3.2 Error Handling

Admin notifications sent via email (d@zborovskiy.com) for:

Message delivery failures

Chat access errors

Bot startup/shutdown events

Configuration loading issues

3.3 Deployment

Platform: Digital Ocean Droplet

OS: Ubuntu 24.10 x64

Dependencies:

Python 3.x

python-telegram-bot library

SMTP service for email notifications

4. Non-Functional Requirements

4.1 Reliability

Bot must automatically recover after crashes

Schedules must persist across restarts

Failed message attempts should be logged and reported

4.2 Security

Bot configuration accessible only through backend

No user commands or interactions accepted

Secure storage of bot token and sensitive data

4.3 Maintainability

Clear logging of all operations

Easy configuration updates through JSON file

Modular code structure for easy modifications

5. Future Considerations

Potential addition of more content types

Possible expansion to support different messages per chat

Monitoring dashboard for bot status