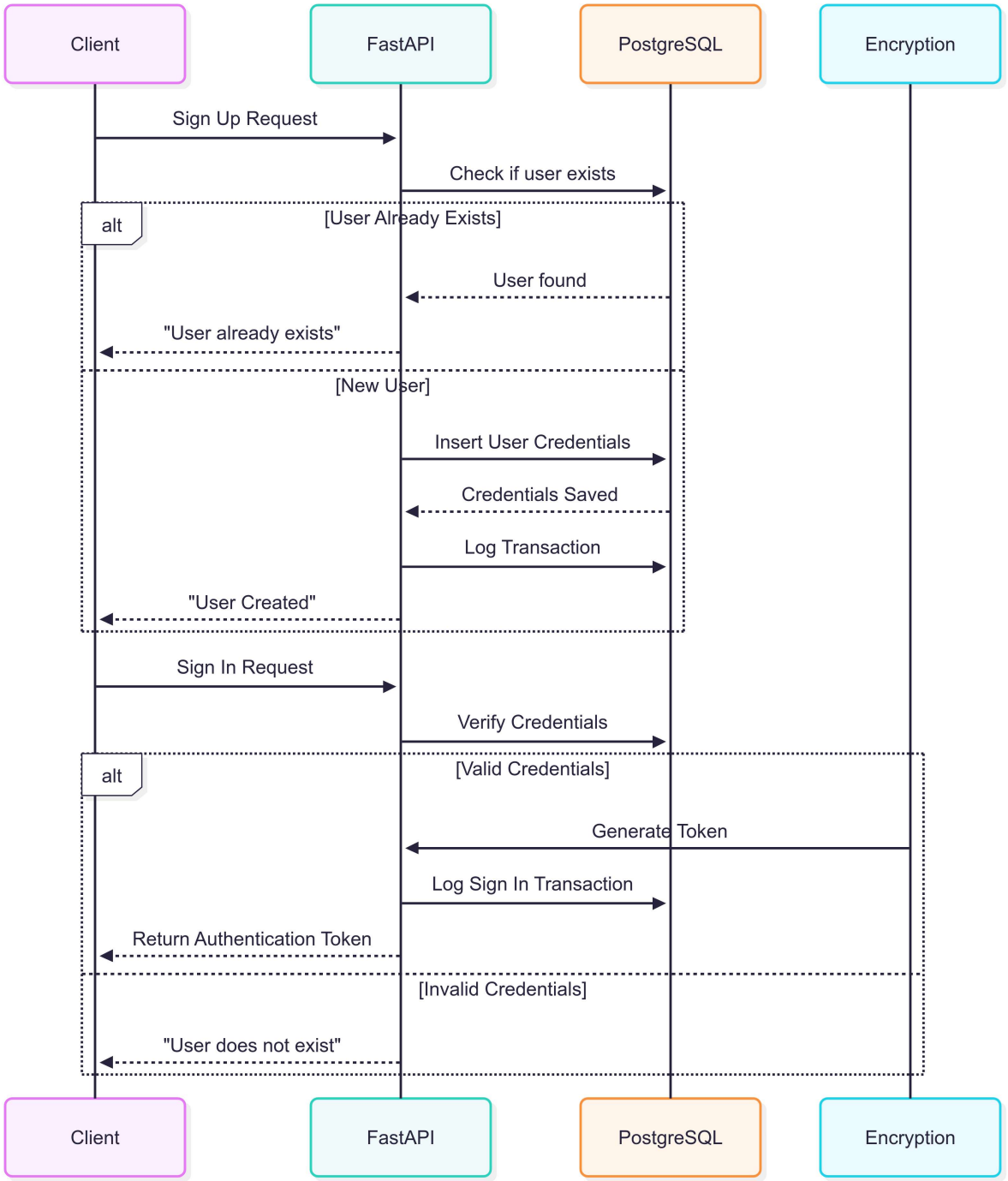
Smartrentals Code Process Visualization

1. Authentication Flow



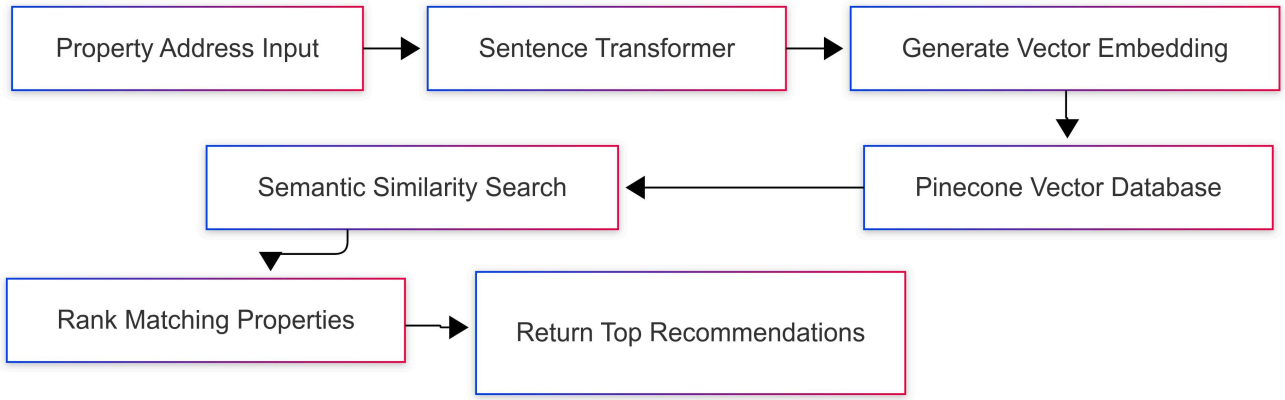
Key Authentication Methods

```python
def generate_token() -> str:
    """Generate a time-based encrypted token"""
    generation_timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S").encode()
    return cipher.encrypt(generation_timestamp)

def validate_token(token_value) -> bool:
    """Validate token based on time constraints"""
    try:
        # Decrypt and check token age
        generation_timestamp = cipher.decrypt(token_value)
        generation_time = datetime.strptime(
            generation_timestamp.decode(),
            "%Y-%m-%d %H:%M:%S"
        )
        current_timestamp = datetime.now()

        # Token valid for only 10 minutes
        return (current_timestamp - generation_time).seconds <= 600
    except:
        return False
```

2. Recommendation Process



Embedding and Recommendation Logic

```python
def get_embeddings(query):
    """Convert text to vector embeddings"""
    return model.encode(query)

def upsert_to_vectordb(property):
    """Store property in vector database"""
    embeddings = get_embeddings(property.Address)
    prepped = [{
        'id': str(uuid4()),
```

```
            'values': embeddings,
            'metadata': {
                'property': json.dumps(property.dict())
            }
        }]
        index.upsert(prepped)

def get_recommendations(pinecone_index, search_term, top_k=10):
    """Retrieve semantically similar properties"""
    embed = get_embeddings([search_term])
    res = pinecone_index.query(
        vector=embed,
        top_k=top_k,
        include_metadata=True
    )
    return res
```

3. Data Model Validation

```
class Property(BaseModel):
    """Strict property validation model"""
    PropertyTypes: Literal[
        '1 Bedroom', '2 Bedroom', '3 Bedroom',
        '4 Bedroom', 'Studio'
    ]
    Security: Literal[
        'Not Applicable', 'Gated Community',
        'Security Guard'
    ]
    # Additional strict type definitions
```

Technical Innovations

1. Semantic Search

- Uses sentence transformers to convert addresses to vector embeddings
- Enables context-aware property recommendations
- Supports flexible, intelligent search capabilities

2. Secure Authentication

- Time-limited, encrypted tokens

- Prevents unauthorized access
- Limits session duration for security

3. Robust Data Validation

- Pydantic models enforce strict data types
- Prevents invalid data entry
- Provides clear contract for API interactions

Performance Optimizations

- Vectorized search using Pinecone
- Efficient embedding generation
- Minimal computational overhead
- Scalable recommendation system

Author : Shriniwas Kulkarni

- **PCCOE 2026 Btech CSE(AIML)**
- **Email:** kshriniwas180205@gmail.com
- **Phone:** +91 [8999883480]
- **GitHub:** github.com/Shriniwas18K