Appendix: Code Snippets

A.1 Python: Train k-NN Model

```
# Python: Train a k-NN model for Indoor Positioning
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor
import joblib

# Load fingerprint dataset
data = pd.read_csv('fingerprints.csv')

# RSSI readings as features, coordinates as labels
X = data.drop(columns=['x', 'y'])
y = data[['x', 'y']]

# Split dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train k-NN
knn = KNeighborsRegressor(n_neighbors=5)
knn.fit(X_train, y_train)

# Save model
joblib.dump(knn, 'knn_model.pkl')
```

A.2 Python: Flask API for Location Prediction

```
# Python: Simple Flask API for Indoor Position Prediction
from flask import Flask, request, jsonify
import joblib
import numpy as np

app = Flask(__name__)
model = joblib.load('knn_model.pkl')

@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json(force=True)
    rssi_vector = np.array(data['rssi']).reshape(1, -1)
    prediction = model.predict(rssi_vector)
    return jsonify({'x': prediction[0][0], 'y': prediction[0][1]})

if __name__ == '__main__':
    app.run(debug=True)
```

A.3 Android Java: BLE Beacon Scanning

```
// Android Java: Scan BLE Beacons and Read RSSI
BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
BluetoothLeScanner bluetoothLeScanner = bluetoothAdapter.getBluetoothLeScanner();

ScanCallback scanCallback = new ScanCallback() {
    @Override
    public void onScanResult(int callbackType, ScanResult result) {
        String beaconId = result.getDevice().getAddress();
        int rssi = result.getRssi();
```

Appendix: Code Snippets

```
        // Process beaconId and rssi
    }
};

public void startScanning() {
    ScanSettings settings = new ScanSettings.Builder()
        .setScanMode(ScanSettings.SCAN_MODE_LOW_LATENCY)
        .build();
    bluetoothLeScanner.startScan(null, settings, scanCallback);
}
```