Appendix A1 – LCM Occupancy Algorithm Outline

Module: LCM Occupancy
This module estimates current and future space occupancy levels, patterns, and anomalies using real-time feeds, lease conditions, and behavioural insights from property usage data.

Key Inputs:

• IoT or Smart Meter Data (desk usage, lighting, temperature sensors, etc.)
• Lease Clauses (permitted usage, subletting terms, co-working restrictions)
• Historical Occupancy Logs (badge-ins, meeting room usage, access logs)
• Tenant Business Type and Headcount Estimates
• Survey Results from Tenants (if available)

Outputs:

• Occupancy Score (0-100%) per building, floor, and unit
• Utilisation Classifications (Underutilised / Efficient / Overcrowded)
• Predicted Future Occupancy Patterns (based on trend detection)
• Compliance Alert (where occupancy breaches lease conditions)

Sample Pseudocode:

```
function calculateOccupancyScore(sensorData, leaseTerms, historyLogs, tenantType):
baselineOccupancy = average(historyLogs.last_3_months) currentUsage =
mean(sensorData.desk_usage) utilisationRatio = currentUsage / baselineOccupancy if
utilisationRatio > 1.2: classification = "Overcrowded" elif utilisationRatio < 0.5:
classification = "Underutilised" else: classification = "Efficient" leaseBreach =
checkLeaseCompliance(leaseTerms, currentUsage) return { "score": round(currentUsage
* 100, 2), "classification": classification, "leaseBreach": leaseBreach }
```

Innovation & Replication Barrier:

This occupancy scoring model incorporates lease condition awareness and contextual tenant behaviour analysis — a combination not generally found in typical BMS (building management systems). This hybrid approach of space analytics and compliance-aware occupancy risk makes replication difficult without access to both proprietary scoring logic and contextual datasets.