

# PART 1 — Agile Requirement Specification (ARS)

(Business + Functional Requirements for Capability Finder)

## 1. Document Control

Field	Value
Module	Capability Finder
System	Competency Tracker
Version	1.0
Owner	Product Owner
Stakeholders	Product, Engineering, QA, Automation
Methodology	Agile Scrum

## 2. Business Objective


The Capability Finder module enables users to identify employees matching specific skill combinations and organizational filters for staffing, analytics, and export purposes.

Primary goals:

- Reduce staffing decision time
- Enable multi-skill AND-based filtering
- Provide exportable talent data
- Support scalable automation validation

## 3. Actors

Actor	Description
Super Admin	Full visibility
Segment Head	Scoped visibility (future enforcement)
Sub-segment Head	Scoped visibility (future enforcement)
Project Manager	View/search
Team Lead	View/search
Team Member	View/search (restricted future scope)

 Current implementation does not enforce RBAC in search (technical debt).

---

## 4. Functional Requirements (FR)

### FR-01: Skill-Based Search (Mandatory)

- User must select at least one skill.
- System shall return employees who possess ALL selected skills (AND logic).
- Proficiency and experience filters apply to selected skills only.

### FR-02: Organizational Filtering

System shall allow filtering by:


- Sub-segment
- Team (cascading from sub-segment)
- Role

Filtering logic shall be cumulative (AND across all filters).

---

### FR-03: Results Display

System shall:

- Display matching employees in tabular format.
  - Show employee name, sub-segment, team, role.
  - Display top 3 skills ordered by:
    1. Proficiency (DESC)
    2. Last used (DESC)
    3. Skill name (ASC)
  - Support client-side sorting by name, sub-segment,  role, team.
- 

### FR-04: Export Functionality

System shall allow:

- Export all search results.
  - Export selected employees only.
  - Download Excel file with filtered dataset.
-

FR-05: Empty & Error States

System shall display:

- “No results yet” (before search)
  - “Ready to search”
  - “No matching talent found”
  - “Search failed”
- 

FR-06: Data Constraints

- Proficiency: 0–5
  - Experience:  $\geq 0$
  - Skills must exist in master
  - Skills with no employees shall be visible but non-selectable
- 

5. Non-Functional Requirements (NFR)

ID	Requirement
NFR-01	Search response < 2 seconds for 1k employees
NFR-02	UI must not freeze for large datasets
NFR-03	Export must complete within 10 seconds
NFR-04	Soft-deleted employees must not appear (expected behavior)
NFR-05	Future RBAC enforcement must be pluggable

↓

---

6. Assumptions

- Search is server-side.
  - Sorting is client-side.
  - No pagination implemented.
  - Soft delete filtering currently missing (known gap).
-

## 7. Out of Scope

- OR logic for skills
  - Saved searches
  - Server-side pagination
  - Allocation % filtering
- 

## 8. Acceptance Criteria (Sample High-Priority)

Example:

### AC-01

Given user selects skills A and B

When search is executed

Then only employees having both skills A and B are returned

### AC-02

Given proficiency filter = 4

When search is executed

Then only employees with proficiency  $\geq 4$  for all selected skills are returned

### AC-03

Given employee is soft deleted

When search is executed

Then employee must not appear in results