University Research Project Management System (URPMS)

Scope

The University Research Project Management System (URPMS) is designed to transform how universities plan, organize, and monitor their research activities. Instead of scattered spreadsheets and disconnected communication, URPMS provides a single, intuitive platform that captures the entire research journey, from proposal drafting and funding acquisition to project execution, dataset management, publication tracking, and final reporting.

What makes URPMS stand out is its focus on the people behind research: Principal Investigators, Co-Investigators, Research Assistants, project managers, finance officers, and external collaborators. Each user experiences tailored functionality, ensuring efficiency while preserving accountability and transparency.

Intended Functionality (User Perspective)

The University Research Project Management System (URPMS) transforms research management into a structured yet intuitive experience by aligning people, projects, tasks, funding, and publications within one integrated environment.

At its foundation is the Person entity, which represents every participant in the system. Persons are classified as Faculty, Staff, or Students, each with role specific attributes such as title, department, or enrolment year. This structure ensures that every individual's position within the research ecosystem is clearly defined.

Research activities are organized through the Project entity, which captures the project's title, description, timeframe, and status. Projects are not isolated records; they are connected hubs. Each project explicitly records who participates, ensuring that collaboration between faculty, staff, and students is transparent.

Projects are supported by the funding entity, where details such as sponsor, amount, and funding period are tracked. This direct funded By relationship creates accountability between financial resources and the projects they enable.

Within each project, work is operationalized through the task entity. Tasks are linked to projects via the has Task relationship and carry essential details including description, due date, status, and priority. Through the assigned To relationship, tasks are mapped directly to responsible persons, ensuring clarity of responsibility and measurable progress.

The system also connects projects to their Publications, documenting outputs with attributes such as title, venue, year, and Digital Object Identifier (DOI). The authors relationship links publications to contributing persons, enabling recognition of individual contributions and tracing the impact of projects through scholarly outputs.

Altogether, URPMS provides a seamless flow, Persons participate in Projects which are funded, decomposed into Tasks, and generate Publications. This structured mapping of entities supports not only day-to-day operations but also strategic oversight. Administrators and researchers alike can instantly trace the lifecycle of research, from funding through assigned work to published results, without ambiguity.

By grounding its design in this ER model, URPMS ensures data integrity, clarity of roles, and transparency of outcomes, while presenting users with a collaborative and goal-oriented research environment.

Entities and Attributes

Person: Core details include Person ID, Name, Email, Affiliation.

ISA hierarchy:

Faculty → Department, Academic Title

Postdoc → Contract End Date

PhD Student → Supervisor, Expected Graduation

Research Project: Defined by Project Code, Title, Summary, StartDate, End Date, Budget Overview, Status.

Funding Grant: Captures Grant Number, Agency, Amount, StartDate, End Date, Restrictions.

Dataset: Stores Dataset ID, Title, Description, Format, Size, Storage Location, Access Level, DOI.

Publication: Includes Pub ID, Title, Venue, Pub Date, DOI, Type.

Task: Attributes include Task ID, Title, Description, Status, Due Date, Priority.

Milestone: Identified by Milestone ID, with Description, Due Date, Achieved Date.

Institution: Records Institution ID, Name, Type (internal/external), Contact.

Relationships

The strength of the model lies in its web of relationships. Each one mirrors a structural rule of the research environment:

Person – works on – Research Project: Many-to-many, with role (PI/Co-I/RA) and allocation% as relationship attributes.

Research Project – funded by – Funding Grant: Many-to-many, reflecting diverse funding structures.

Research Project – produces – Dataset: One-to-many; a project can generate several datasets.

Research Project – produces – Publication: Many-to-many; publications can result from multiple projects, and projects can yield multiple outputs.

Task – assigned to – Person: Many-to-many; tasks may involve multiple collaborators.

Milestone – part of – Research Project: One-to-many; each milestone belongs to a single project.

Person – affiliated with – Institution: Many-to-one; individuals belong to one institution.

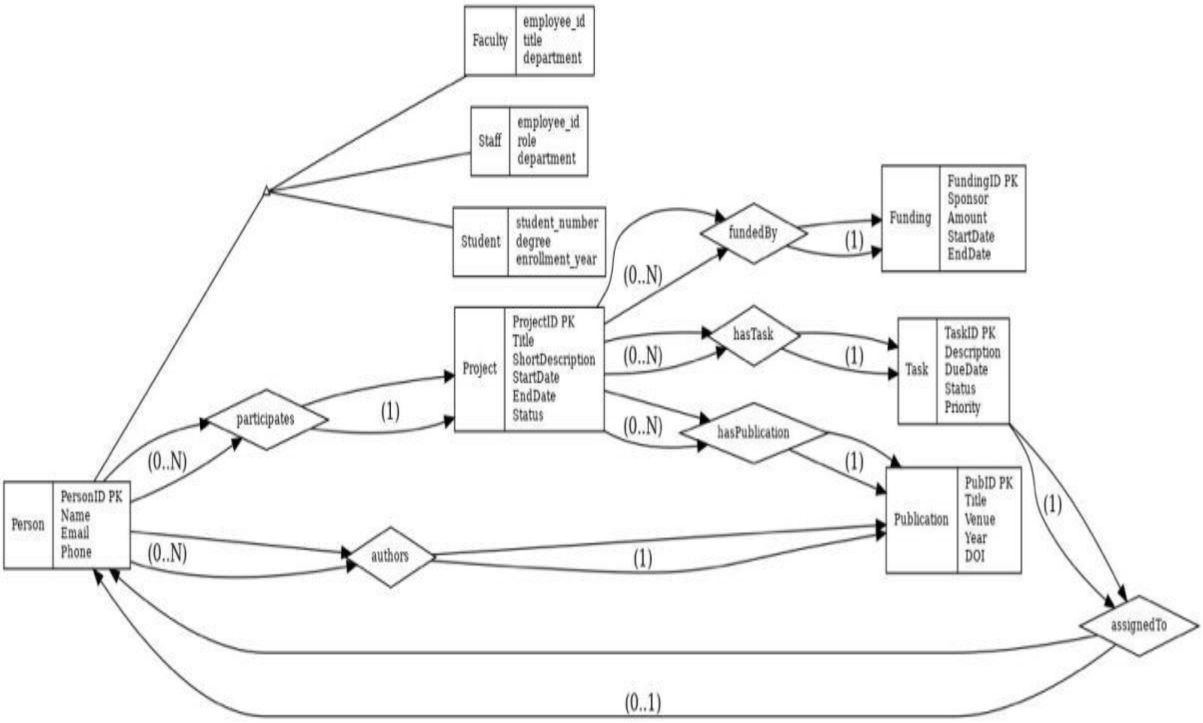
Research Project – collaborates with – Institution: Many-to-many; collaboration across departments or universities.

Cardinalities such as one-to-many (1:M) and many-to-many (M: N) specify the real-world rules that govern associations between entities. For example, a *Research Project* can generate multiple *Tasks* or *Datasets* (1:M), while a *Person* may participate in several projects and each project may involve many people (M: N). These many-to-many cases are represented through associative entities such as *Participation* and *Authorship*, which also record additional details like roles, order of contribution, or duration of involvement.

By modelling both 1:M and M: N relationships, the URPMS data model ensures:

- **Realism** → capturing the inherently collaborative nature of academic research.
- **Precision** → avoiding ambiguity and ensuring clear, structured links between entities.
- **Extensibility** → allowing future features, such as patents or ethics approvals, to be added without disrupting the existing model.

ER Diagram



User Interactions:

What the user *sees* (UI / visible data)

- **Dashboard / Project list:** project titles, status, timeframe, budget overview, and quick links to project pages (implied by the Project entity attributes).
- **Person profiles:** name, email, affiliation, role-specific attributes (e.g., Faculty → department/title; PhD student → supervisor, expected graduation).
- **Project detail pages:** summary/description, start/end dates, status, attached funding grants, tasks, milestones, datasets and linked publications.
- **Task views:** task title, description, status, due date, priority and assigned persons.
- **Funding/grant pages:** grant number, agency, amount, active dates, and any restrictions.
- **Dataset / Publication records:** metadata such as dataset DOI, format, size, storage location; publication DOI, venue, year and type.

Typical user actions (UI actions they can take) and immediate results

- **Create / edit a Person profile**
  Result: New or updated person entry (with role-specific fields) and association to an Institution.
- **Create / edit a Research Project**
  Result: Project record with title, summary, dates, budget and status; becomes a hub that can link to persons, tasks, funding, datasets, publications.
- **Assign people to a project (Participation)**
  Action: add a person to a project with role (PI/Co-I/RA) and allocation%
  Result: Participation associative record; project membership and role become query able/auditable.
- **Link funding grants to projects**
  Action: associate one or more Funding Grant records with a project
  Result: funded By relationship created; finances become traceable to projects.
- **Create / assign / update Tasks and Milestones**
  Result: Tasks get assigned To persons (many-to-many allowed) and can be tracked by status/priority/due date; milestones link to projects.
- **Upload / register Datasets**
  Action: add dataset metadata (format, size, storage, DOI, access level) to a project
  Result: dataset record created and linked to the producing project; access level controls who can use it.

- **Register Publications / Authorship**

  Action: add publications and link authors (authorship associative entity)

  Result: publications become connected to contributing people and projects; author order/roles can be captured.
- **Search / report / trace relationships**

  Action: query projects → see funding, participants, tasks, datasets, publications

  Result: traceability from funding → tasks → outputs, enabling oversight.

System behavior implied by the data model (important consequences for users)

- **Many-to-many relationships are explicit and carry attributes** (e.g., Participation, Authorship) — so when a user links a person to a project or a publication to several projects, the system stores role, allocation% or author order, not just a simple link.
- **Access control is implied for datasets** (the model mentions Access Level), so users will see or not see datasets depending on permissions.

Illegal input/actions

□ **Unauthorized access / privilege abuse** — logging in as another user, elevating privileges, or accessing data beyond your role (e.g., viewing restricted datasets or financial details without permission).

□ **Uploading restricted or copyrighted material without rights** — adding datasets or files to which the uploader does not hold the necessary rights or that violate licence/ethics agreements.

□ **Tampering with audit/associative data** — changing participation/allocation or authorship order fraudulently to misrepresent contributions.

□ **Inserting malformed or malicious inputs** — e.g., attempts at SQL injection, uploading malware, or submitting malformed metadata that breaks processing.

□ **Bypassing funding restrictions** — misassigning grant funds to projects in violation of recorded grant restrictions.

□ **Privacy breaches** — exposing personally identifiable information (PII) improperly or exporting confidential review content.

Short checklist for an implementer / admin (practical next steps)

□ Make clear UI affordances for creating/editing Person, Project, Funding, Task, Dataset, Publication.

□ Enforce role-based permissions (Faculty, Staff, Student; PI/Co-I/RA roles).

□ Add validations for DOIs, dates (StartDate ≤ EndDate), numeric budget fields, and allocation% totals.