# Developing Effective Requirements

Software Analysis and Design

Class 08

Mustafa Hasan

# From System to User

## What System Shall Do

- **System shall** auto-populate student's schedule based on registered classes from the university's academic calendar

## What Users Want

- **As a student, I want to** see the schedule of my classes so that I may know when I can register for other events


Dean

I would like to know why students do not attend events?

Students

I do not know the event schedule

I forget about them

They often conflict with my class schedule!

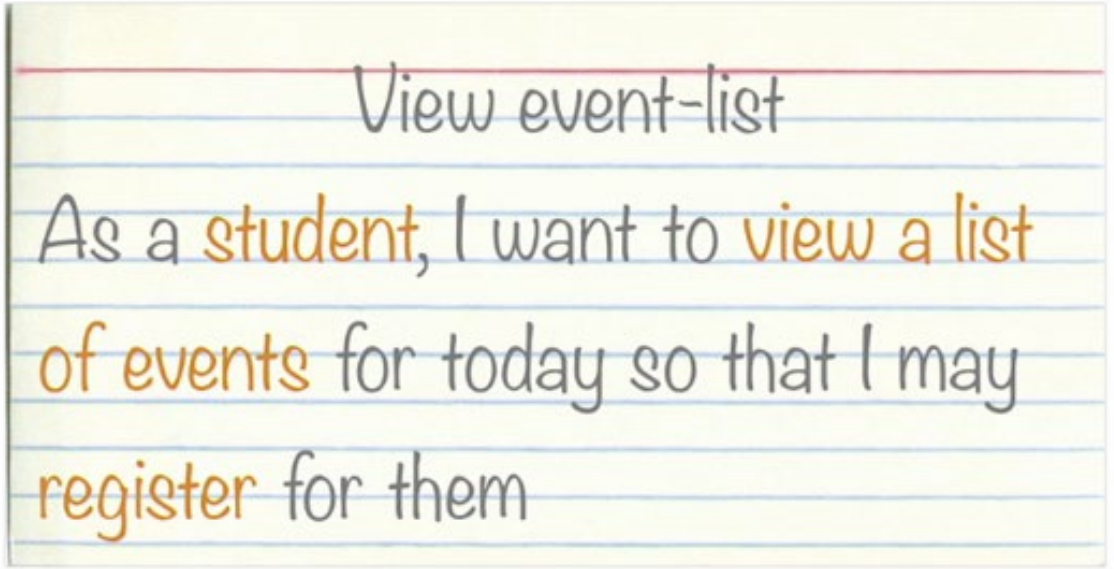
# User story: a requirement articulated from a user's perspective

## View event-list

As a student, I want to view a list of events for today so that I may register for them

| Manage event schedules | View events | Register for events | Generate alerts | Generate reports |
|---|---|---|---|---|
| As a scheduler, I want to see all open time slots in a time period | As a student, I want to view the events so that I may register for them | As a student, I want to see when registering for an event if it conflicts with another registered event | As a student, I want to receive alerts to let me know when new events are scheduled | As a scheduler, I want to see all open time slots in a time period |
| As a scheduler, I want to see the available venues for scheduling an event | As a student, I want to see how many other students have registered for an event | | As a scheduler, I want to receive an alert if another event is scheduled that conflicts with my event | |
| As a scheduler, I want to see the available venues for scheduling an event | | | As a student, I want to receive an alert if event schedule is changed | |
| As a scheduler, I want to schedule an event in a time slot | | | | |

# User Story Template




As a <user-type>, I want <to do a task> so that <a goal>.

Who What Why

# Conditions of satisfaction: acceptance criteria for confirmation.

## View event-list

As a student, I want to view a list of events for today so that **I may register for them**

## Conditions of satisfaction

- Verify that all events are listed
- Verify that events registered for are marked so
- Verify that conflicts are shown


1

## Epic

High-level functionalities

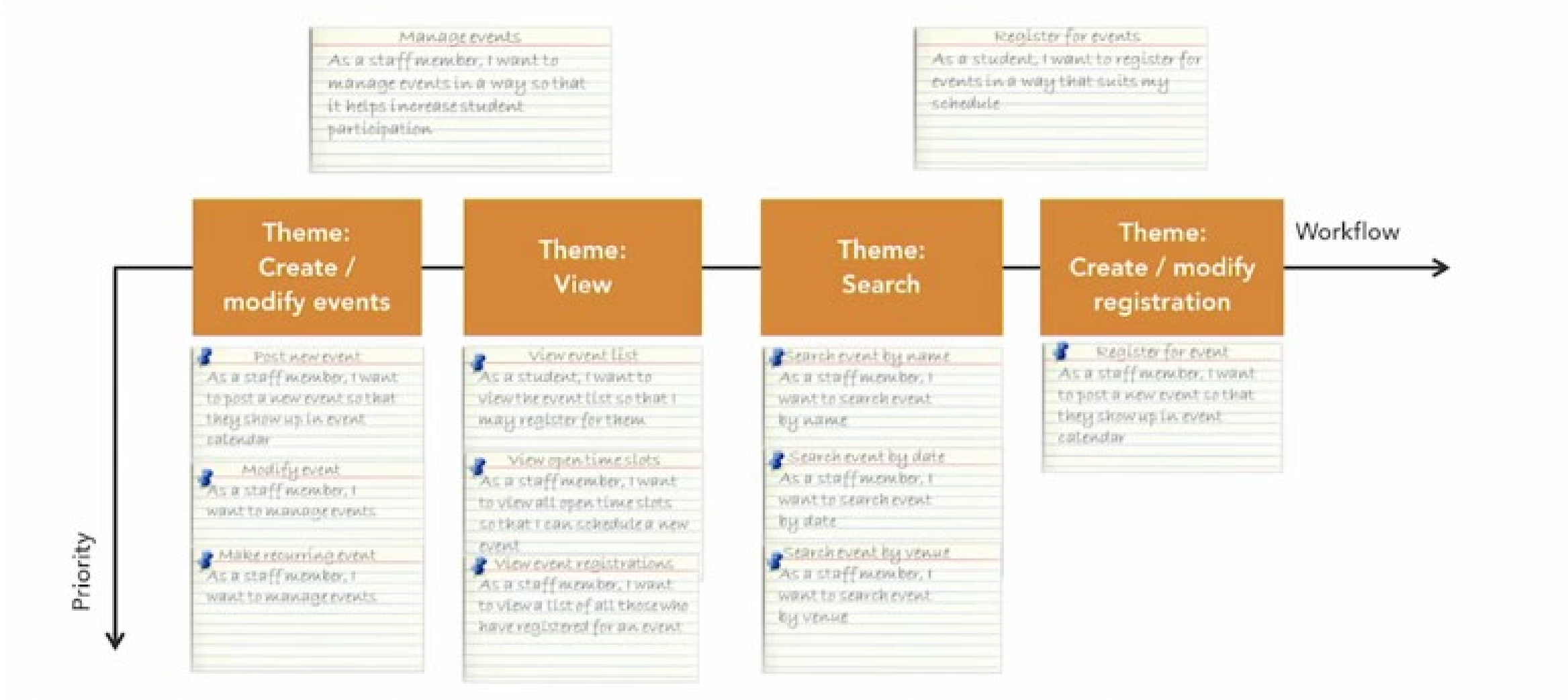


2

## Theme

Collection of related stories

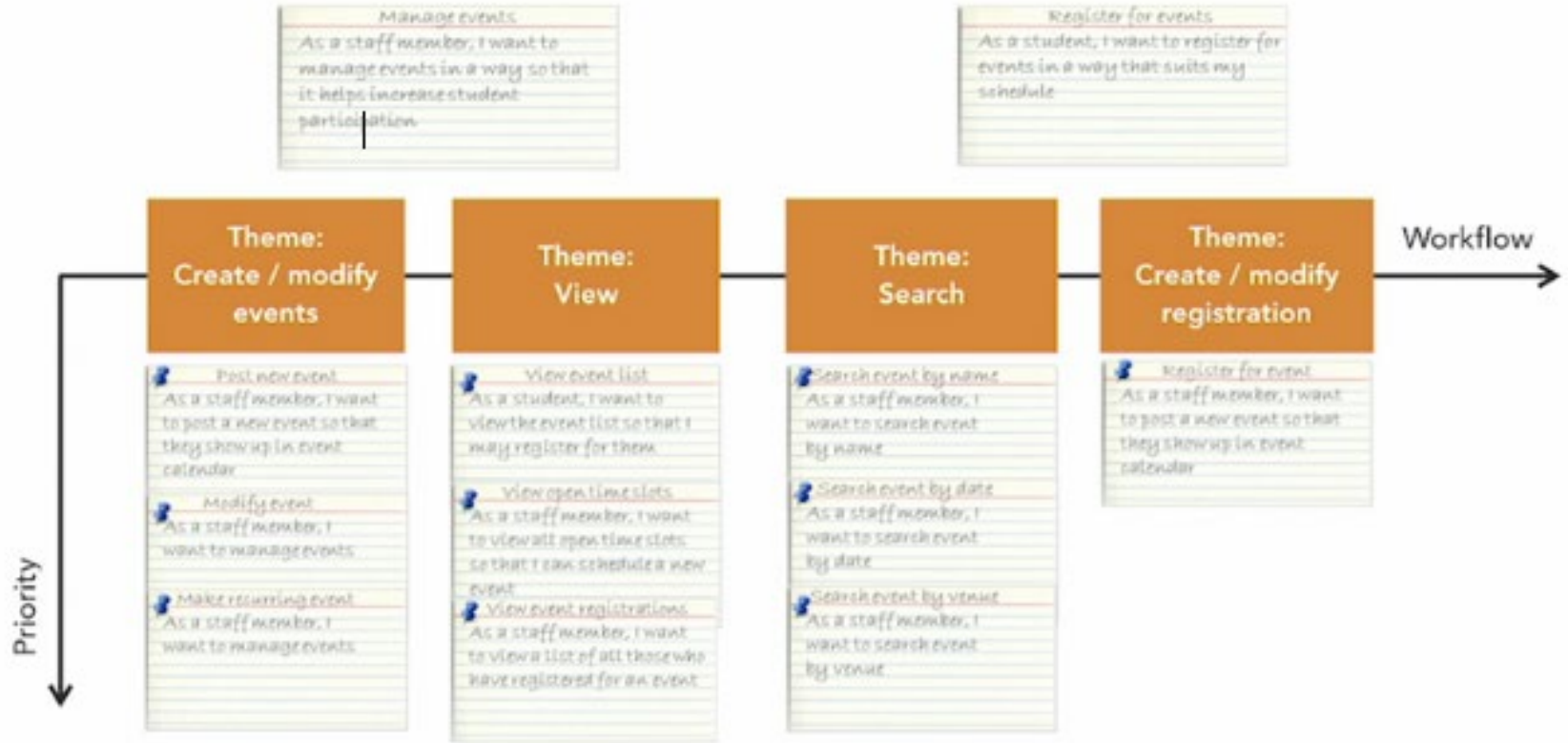
# Managing user stories through story mapping



# Story Mapping

## User stories

- Decomposed
- Sequenced
- Themed
- Prioritized



1

## User Story

A short description of a requirement from the user's perspective

As a <user-type> I want to <task> so that <goal>

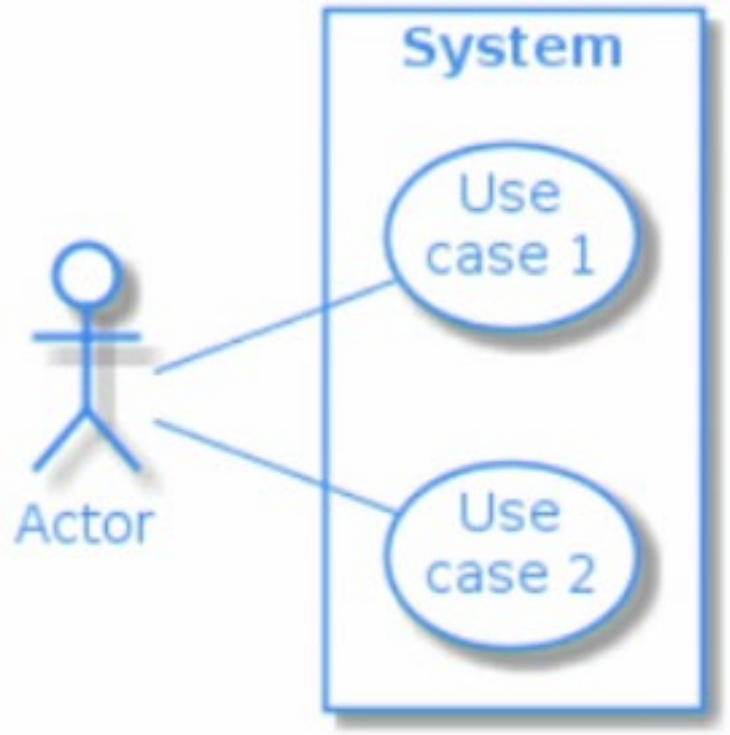
2

## Use Case

An interaction between the system and an actor through which the actor achieves some value

# Use Case Template

- Title
- Primary actors
- Secondary actors
- Preconditions
- Basic flow



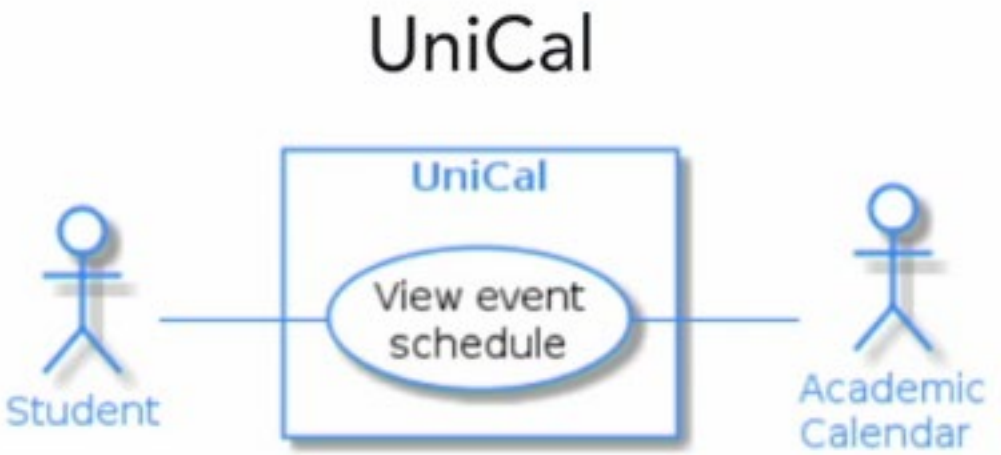
# Example

- Title: view event schedule
- Primary actor: student
- Secondary actor: academic calendar

# Example

## Precondition

- UniCal application is open
- Student is logged into UniCal
- UniCal's option to view schedule is visible



# Example(Cont)

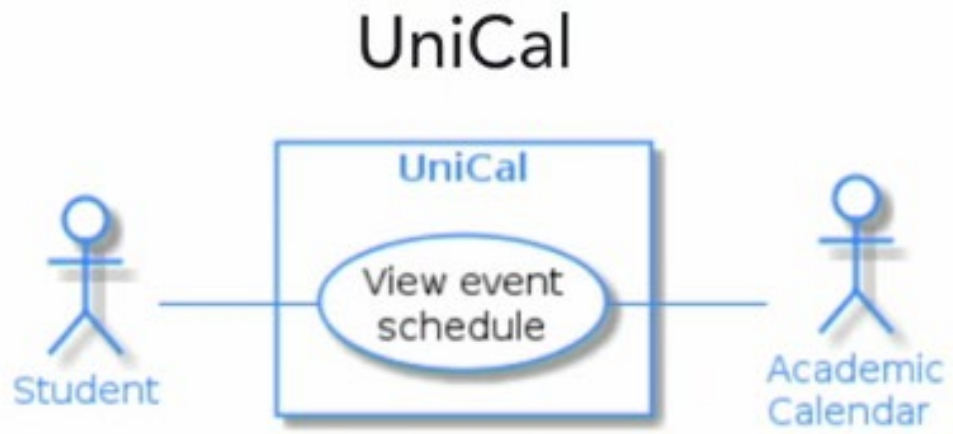
**Basic flow**

1. Student chooses the option to view event schedule.
2. UniCal gives options to choose the period for event schedule.
3. Student chooses the period.

# Example(Cont)

**Basic flow**

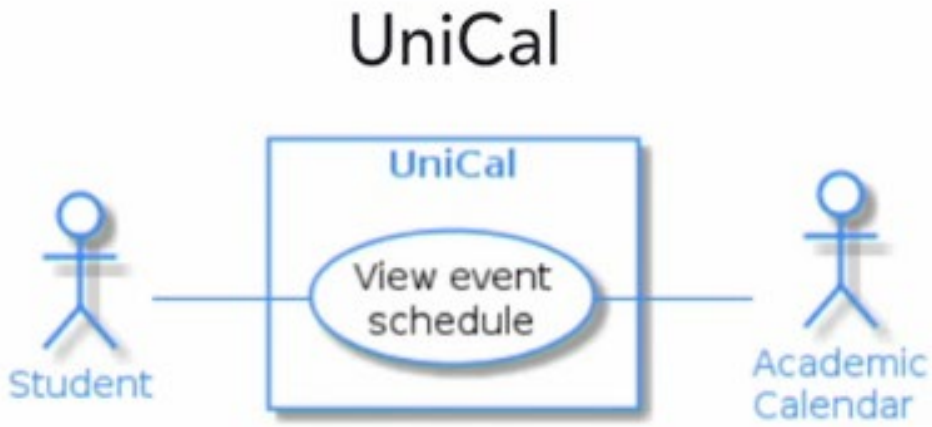
4. UniCal displays the list of events scheduled in the period with an option to exit the view.
5. Student views the schedule and exits.
6. Use case ends successfully.



# Example (Cont)

## Alternate flow

1. After step 2 in basic flow, student selects period for which no events are scheduled.
2. UniCal displays a message to convey the above with an option to exit.
3. Student exits.



# Example (Cont)

**Postcondition**

- UniCal is back to its original open state

# Use Case vs. User Story

## Use Case

- Originated in unified process methodology
- Represents a user interaction
- Requirements analyzed and captured in a document

## User Story

- Originated as part of scrum framework
- Represents a user need
- Requirements understood through conversations

# Use Case vs. User Story

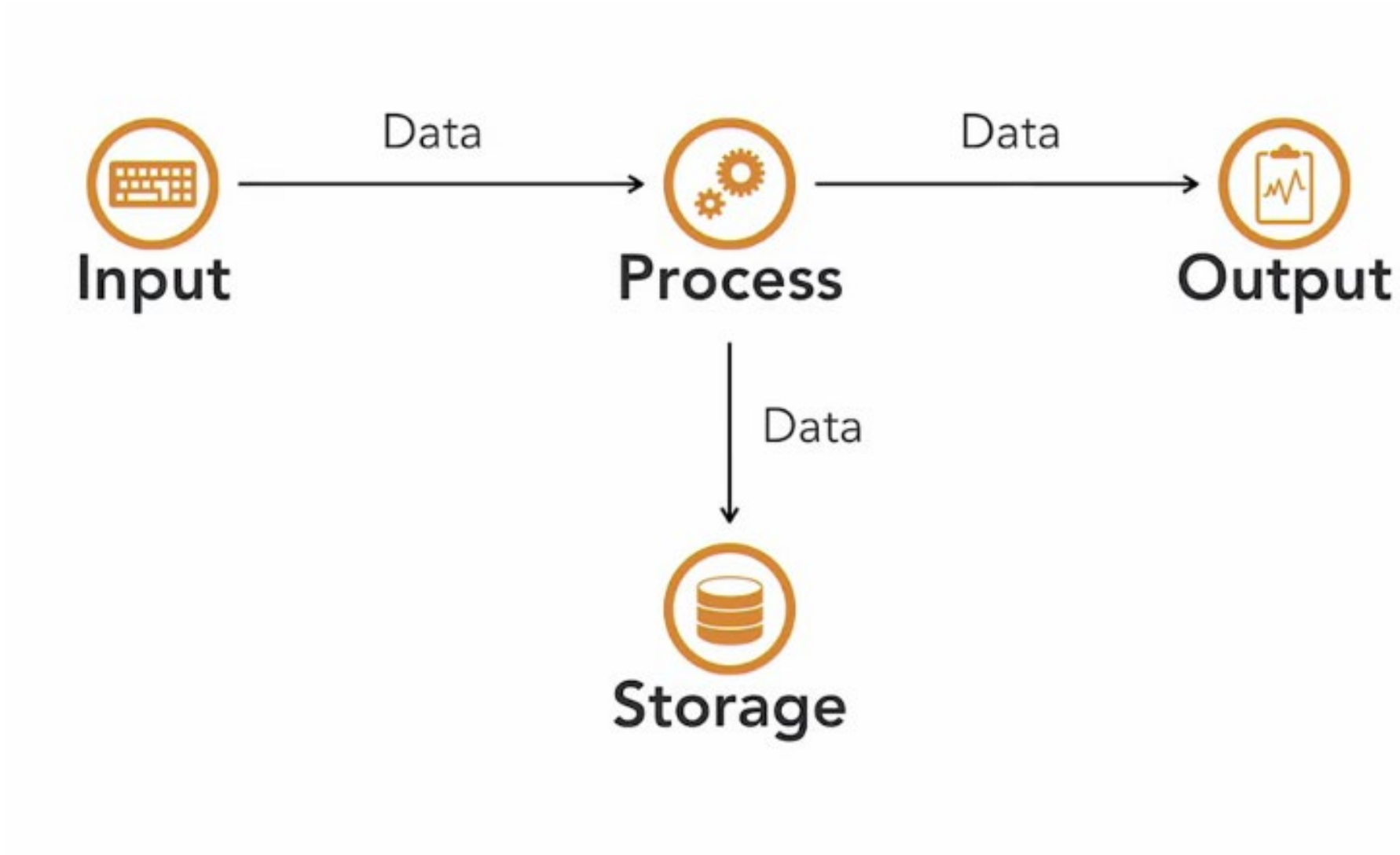
## Use Case

- Suitable for projects with complex requirements and large distributed teams

## User Story

- Suitable for projects with small collocated teams and frequent access to the user

# Data Requirements

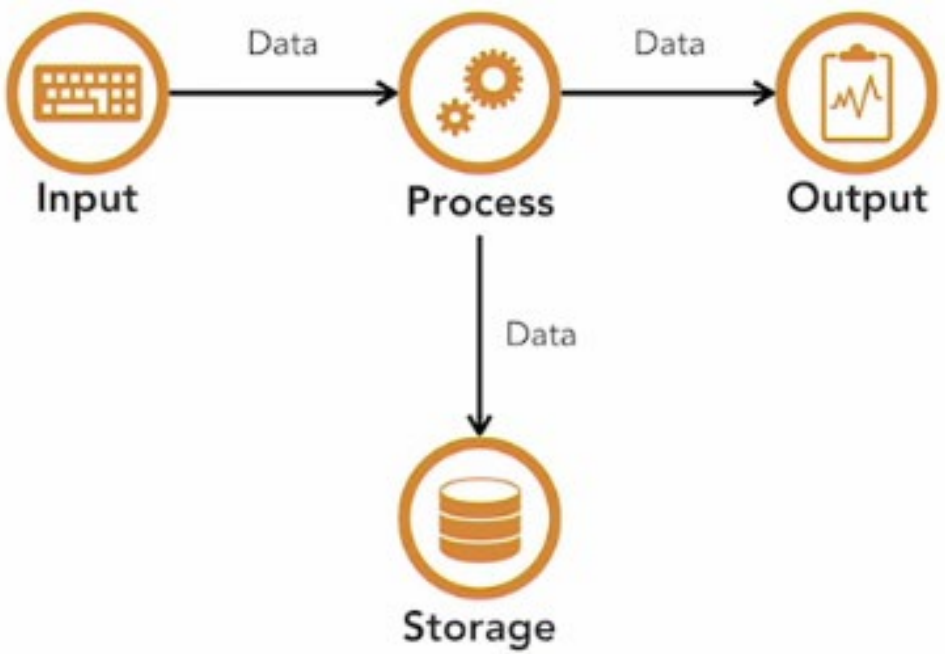


# Data Characteristics

**Functional meaning**

**Nonfunctional properties**

- Source
- Destination
- Format
- Frequency

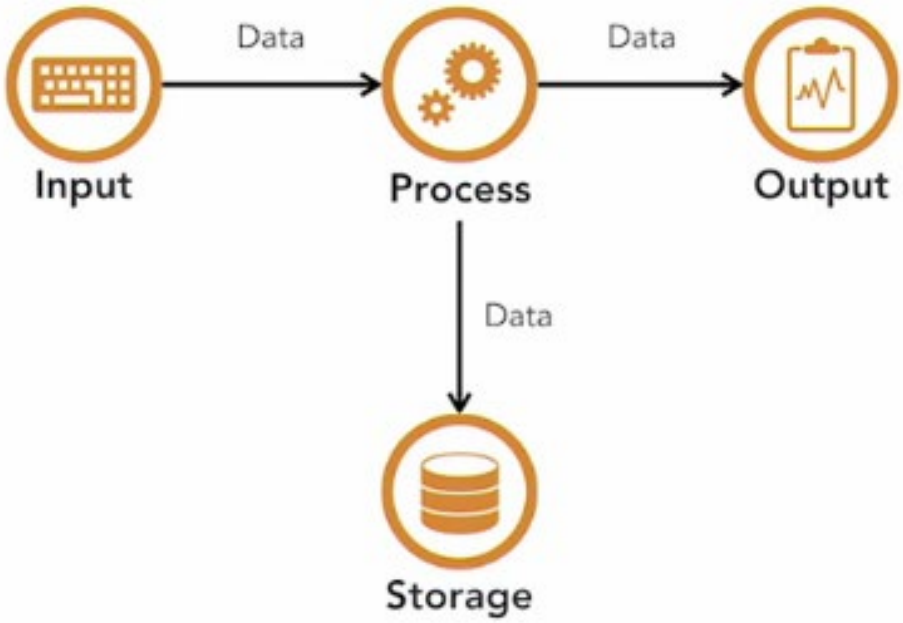


# Data Characteristics

**Functional meaning**

**Nonfunctional properties**

- Volume
- Volatility
- Integrity
- Consistency



# Data represents domain knowledge

| Assets | Liabilities | Equity |
| --- | --- | --- |
| **Current assets**<br>• Cash/bank<br>• Receivable<br>• Inventory<br>• Prepaid expenses<br>**Fixed assets**<br>• Land<br>• Building | **Short term**<br>• Payables<br>**Long term**<br>• Loans | • Retained earnings<br>• Shareholders equity |

**Assets = Liabilities + Equity**

**Event**
- Seminar
- Conference
- Lecture
- Celebration

**Members**
- Employee
- Student
- Visitors

**Event Schedule**
- Date
- Time
- …

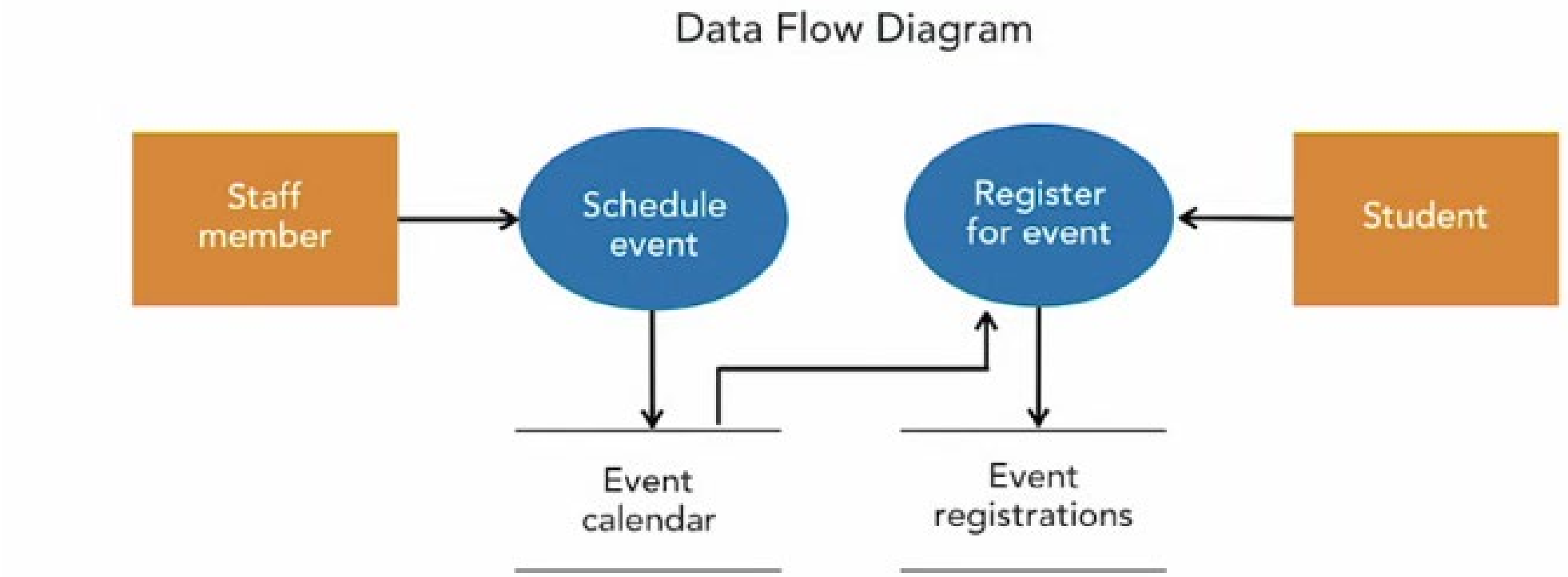
**Data Dictionary**

| Data element | Description | Composition or data type | Length | Values |
|---|---|---|---|---|
| Event | A scheduled activity | Event title, event type, date, time, duration, scheduler | | |
| Event type | Classifier for expected audience size, AV facilities | String | 100 | Seminar, conference, lecture, celebration |
| Location | Event venue | Campus, building, room number | | |
| … | | | | |

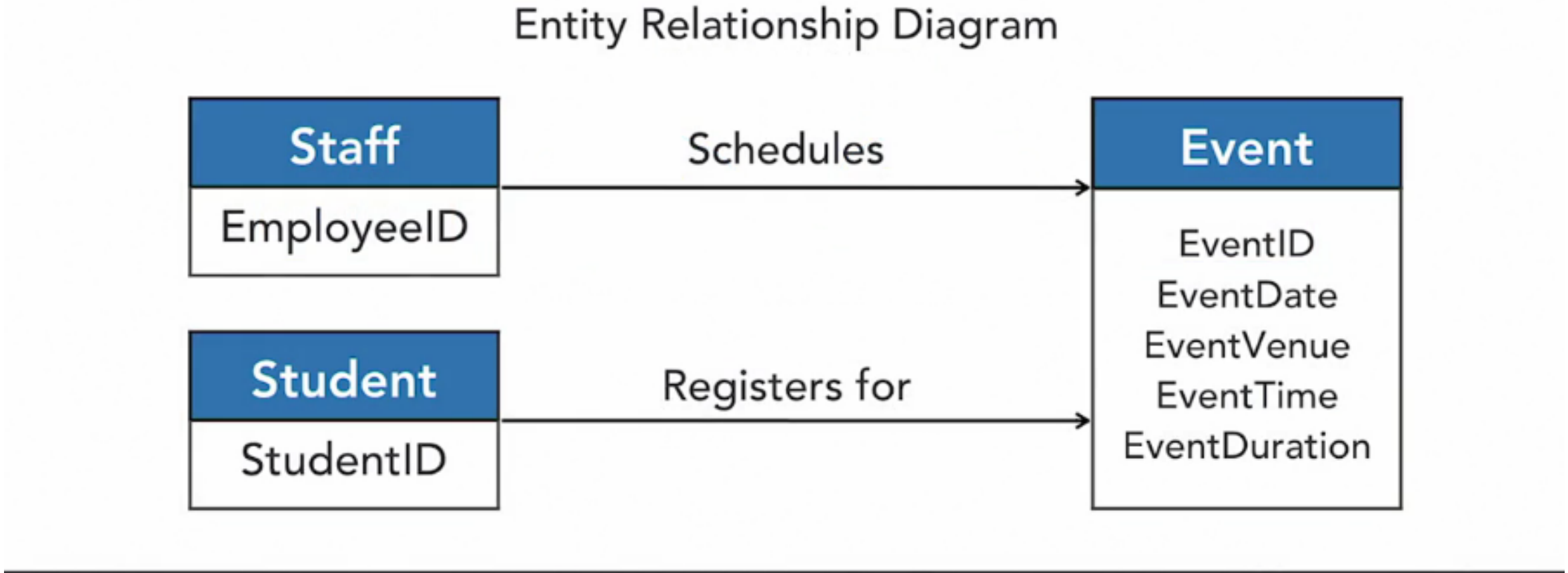
A data dictionary is a collection of detailed information about the data entities used in an application.

| CRUD Matrix | | | |
|---|---|---|---|
| Use case | Calendar | Event | Registration |
| Schedule event | R | CRUD | |
| Edit event | R | U | |
| Register for event | R | R | CRUD |
| View open slots | R | R | |
| View conflicts | R | R | R |
| … | | | |

CRUD matrix correlates use cases with “create, read, update, and delete” access to data entities.




**Data flow diagrams (DFDs) provide a big-picture view of how data moves through a system.**



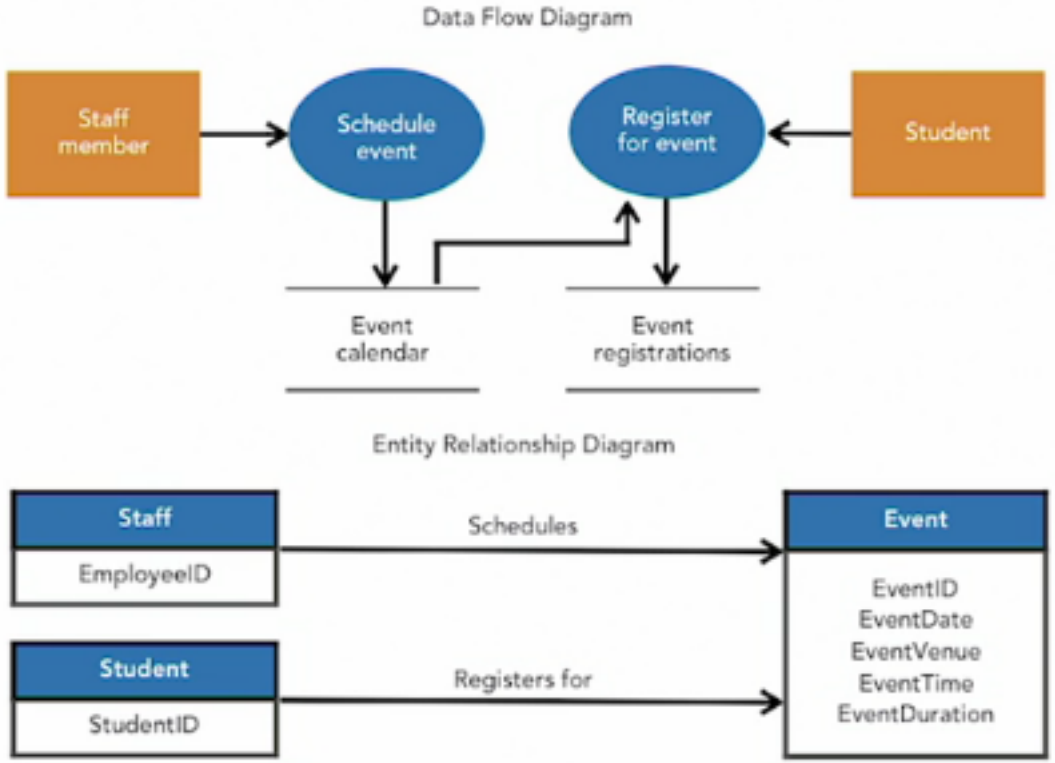

Entity relationship diagrams (ERDs) used in requirements analysis focus on relationships among data entities irrespective of their implementation in a database

**Data Dictionary**

| Data element | Description | Composition or data type | Length | Values |
|---|---|---|---|---|
| Event | | | | |
| Event type | | | | |
| Location | | | | |
| … | | | | |

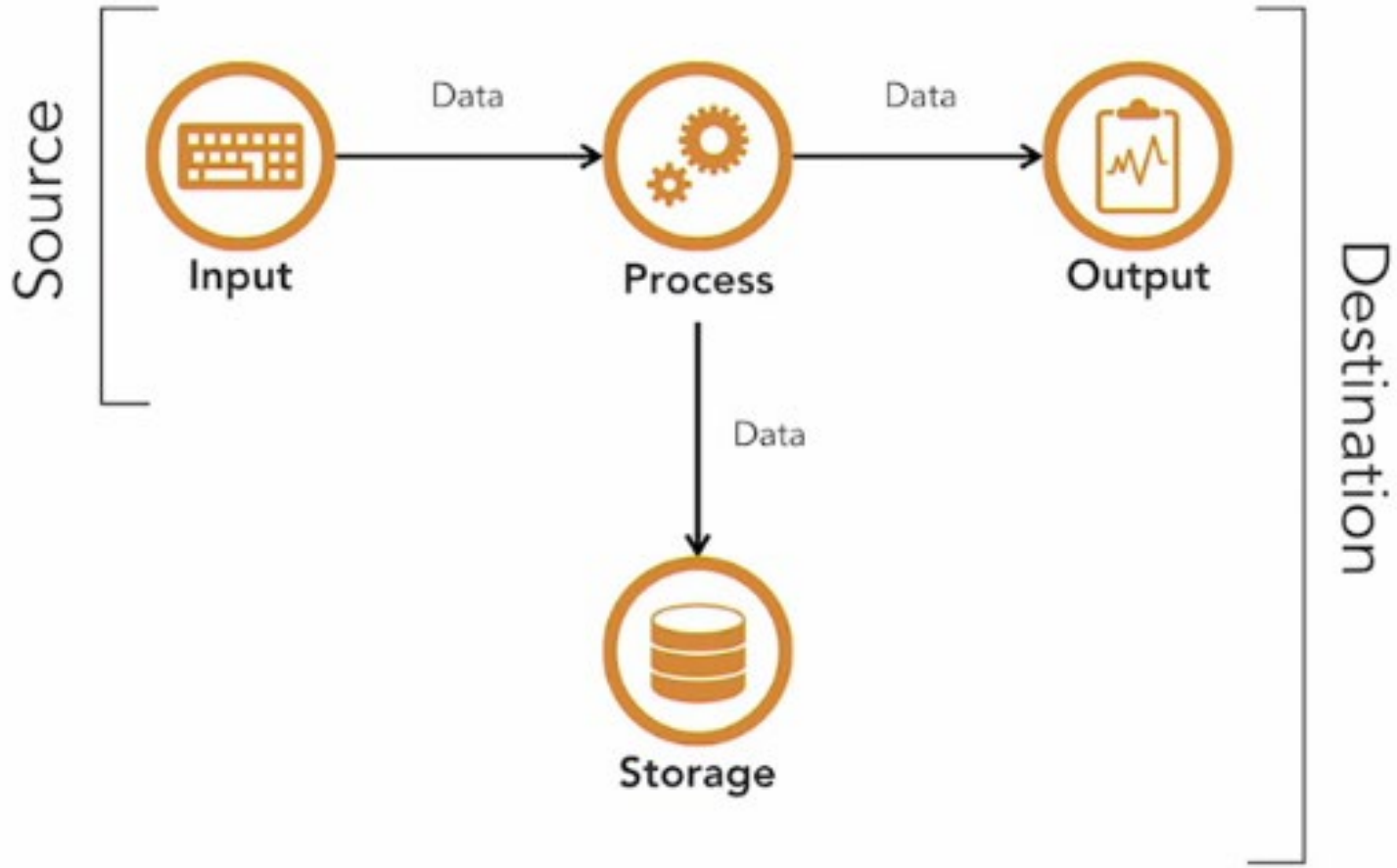
**CRUD Matrix**

| Use case | Calendar | Event | Registration |
|---|---|---|---|
| Schedule event | R | CRUD | |
| Edit event | R | U | |
| Register for event | R | R | CRUD |
| View open slots | R | R | |
| View conflicts | R | R | R |
| … | | | |



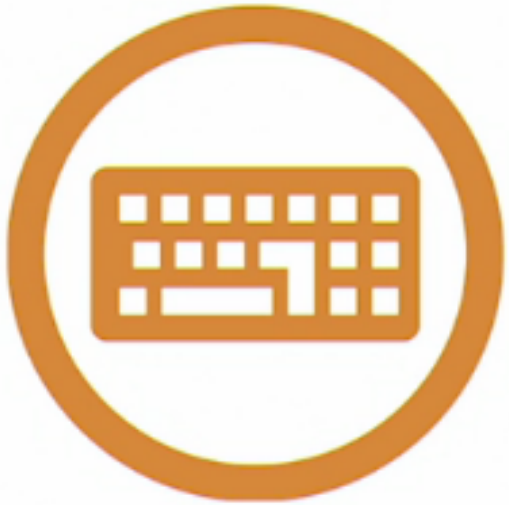

# Understanding data content helps in functional requirements analysis.

Nonfunctional perspective focuses on structural properties of source data and destination data.



# Source

- Text: coding and language
- Audio/video: streaming or download file format
- Signal: analog or digital
- Format, integrity, consistency: need for ETL (extract, transform, load) functionality



# Source

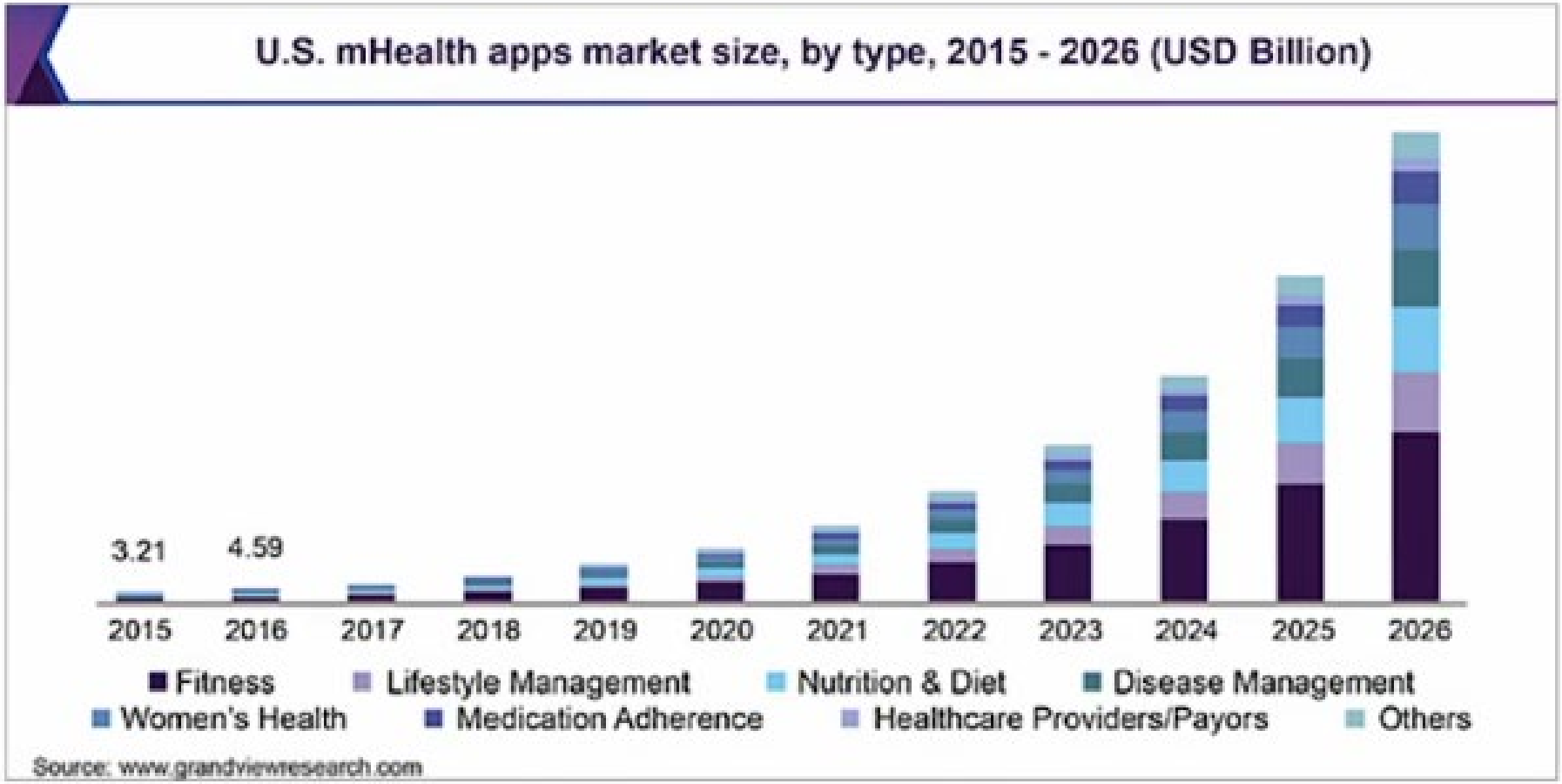
- Volume: bandwidth and storage
- Volatility: database performance

# Destination

**Report:**

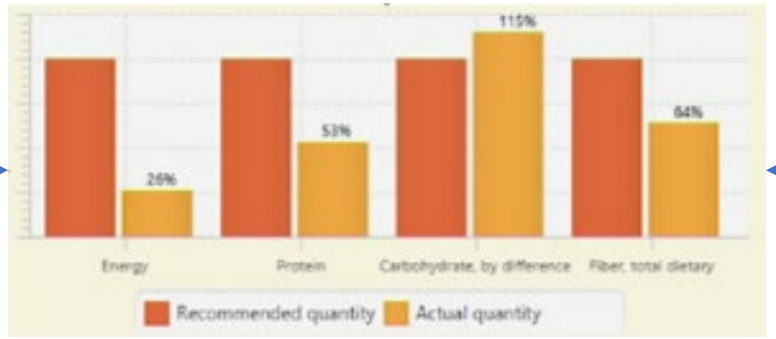
- Latency

# Class Work (Challenge 3) One Use Case and One User Story



# Nutrition Tracking

PAU Health Inc. It targets those who prefer to cook at home, but don't have the tools or the information to assess the nutritional quality of what they cook.




PAU Health Inc. wants your team to develop a nutrition tracking app, using which users can track their daily nutrition in terms of calories, protein, fat, and carbohydrates

BSTI Have the Data about all food in Bangladesh

# Nutrition Tracking Requirements

1. Track nutrition by meal.
2. Take nutrition data for all food products from BSTI Bangladesh
3. Suggest nutrition requirements based on user's gender, age, height, and weight.
4. Post a disclaimer on all suggestion outputs that it is based on BSTI recommendations.
5. Achieve minimum 80% adoption among PAU Health Inc customers within on year of product launch.

# Nutrition Tracking Requirements

6. Collect data to track frequency and patterns of usage of all features.
7. First release of the app targeted for two months from start
8. User must log in to access personal data
9. The user data must be protected as per the health Insurance portability and accountability
10. The app should be intuitive and easy to use with no training requirements.