

Assignment - 1

1. 1. Causes of Software Crisis:

1. Scope Creep: Uncontrolled changes in requirements caused delays and inconsistencies.
2. Poor User Interface: The system was not user-friendly, reducing adoption.
3. Integration Issues: Failure to connect with legacy systems led to operational breakdown.

2. Model Evaluation:

- Waterfall Model is rigid & unsuitable for evolving requirements.
- Spiral Model allows iterative development, risk analysis & better handling of integration.

Best choice: Spiral Model due to flexibility & risk management.

3. Process Improvements:

- Use Agile principles for adaptability.
- Apply TSP/ PSP for disciplined process control.
- Conduct early prototyping to validate requirements.

2. 1. Prototype Model:

Helps in gathering user feedback quickly, but lacks strong structure for regulatory compliance and large-scale integration.

Evolutionary Model:

Enables phased delivery of modules, ensures compliance checks at each stage and incorporates continuous stakeholder feedback.

2. The Evolutionary Model is more suitable because it supports gradual integration of modules, meets healthcare regulations, adapts to multiple revisions.

3. Early Requirement Elicitation: Conduct interviews and workshops with doctors, staff and administrators.

Risk Handling: Deliver the system incrementally, testing compliance and integration risks at every stage.

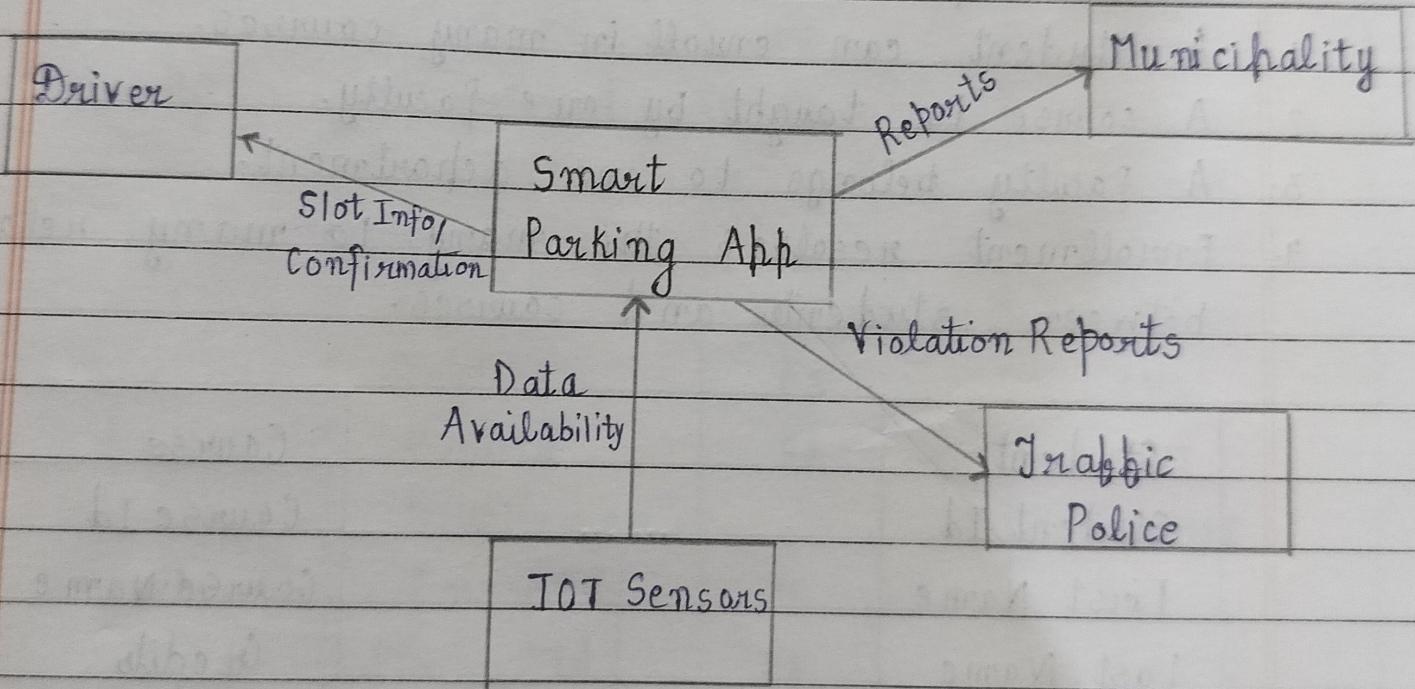
3. 1. Requirement Elicitation:

- Municipality: Interviews and workshops (policy & infrastructure needs).
- Drivers: Surveys and prototypes (usability & feature expectations).

- Traffic Police: Questionnaires and observation (monitoring & enforcement).

2. Context - Level DFD:

- Entities: Driver, Municipality, TOT Sensors, Police.
- System: Smart Parking App.
- Data Flows: Availability data (sensors → app), booking / payment info (driver ↔ app), reports (app → police / municipality).



3. Functional Requirements (FR):

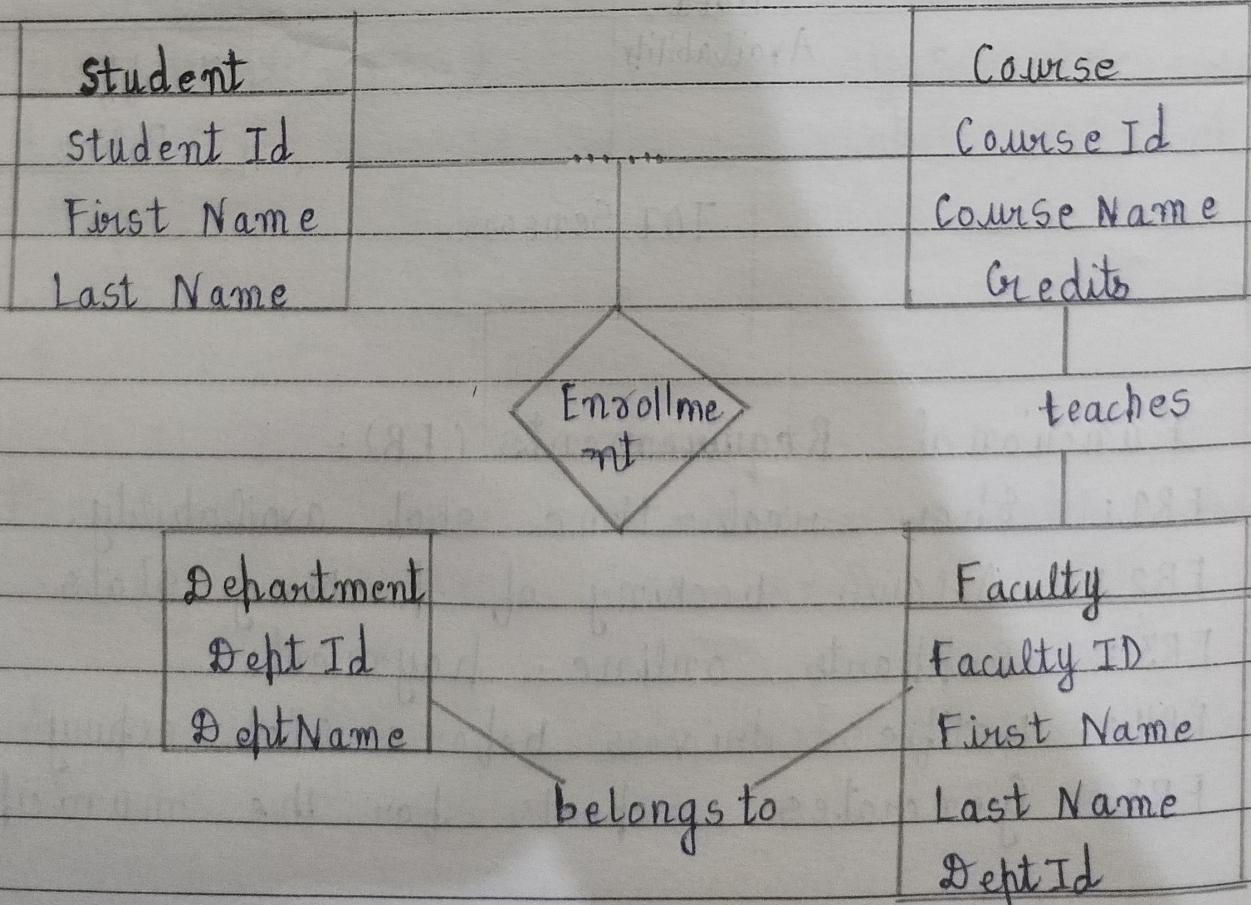
1. FR1: Show real-time slot availability.
2. FR2: Allows booking of parking slots.
3. FR3: Supports online payment.
4. FR4: Notifies drivers before slot expiry.
5. FR5: Generates reports for the municipality.

Non-Functional Requirements (NFR):

1. NFR1: Runs 24/7 without downtime.
2. NFR2: Ensures secure data and payments.
3. NFR3: Responds within two seconds.

4. 1 ER Diagram:

- Entities: Student, Course, Faculty, Department, Enrollment.
- Relationships:
 1. A student can enroll in many courses.
 2. A course is taught by one faculty.
 3. A faculty belongs to one department.
 4. Enrollment resolves the many-to-many relation between students and courses.



2. Data Dictionary Table:

Entity	Attributes	Key	Data Type
Student	StudentId	PK	INT
	First Name		VARCHAR
	Last Name		VARCHAR
	Email	unique	VARCHAR
	DOB		DATE
	DeptId		INT
Course	CourseId	PK	INT
	CourseName		VARCHAR
	Credits		INT
	FacultyId	FR	INT
Faculty	FacultyId	PK	INT
	FacultyName		VARCHAR
	Last Name		VARCHAR
	Email	Unique	VARCHAR
	Dept Id	FK	INT
Department	DeptId	PK	INT
	DeptName		VARCHAR
Enrollment	EnrollId	PK	INT
	StudentId	FK	INT
	CourseId	FK	INT
	Semester		VARCHAR
	Grade		VARCHAR