EXPERIMENT 3

Online Course Reservation System
Data Dictionary

1.User Table:

• ***User_id*:** Integer, Unique identifier for each user (Primary Key, Auto Increment)

• ***Name*:** String (max 100 chars), Full name of the user (Required)

• ***email*:** String (max 100 chars), Unique email address (Rewuired)

• ***password*:** String (Encrypted), Stores user password (Required)

2. Course Table:

• ***course_id*:** Integer, Unique identifier for each course (Primary Key, Auto Incremnt)

• ***course_name*:** String (max 100 chars), Name of the course (Required)

• ***description*:** Text, Detailed course description (Optional)

• ***instructor_id*:** Integer, References instructor (Foreign Key to user_id)

• ***seats_available*:** Int, Number of available seats (Required).

3. Enrollment Table:

• ***enrollment_id*:** Integer, Unique identifier for each enrollment (Primary Key, Auto Increment)

• ***user_id*:** Integer, References the student enrolling (Foreign Key to user_id)

• ***course_id*:** Integer, References the course (Foreign Key to course_id)

• ***enrollment_date*:** DateTime, Date and time of enrollment (Default: Current Time).

4. Registration Process Table:

• ***registration_id*:** Integer, Unique identifier for registration (Primary Key, Auto Increment)

• ***user_id*:** Integer, References student registering (Foreign Key to user_id)

• ***course_id*:** Integer, References the selected course (Foreign Key to course_id)

• ***status*:** Enum ('Pending', 'Completed'), Status of registration (Required).

5. Course Catalog Table:

• ***catalog_id*:** Integer, Unique identifier for catalog entry (Primary Key, Auto Increment)

• ***course_id*:** Integer, References course (Foreign Key to course_id)

• ***availability*:** Boolean, Indicates if seats are available (Required)

6. Payment Table

• ***payment_id*:** Integer, Unique identifier for each payment (Primary Key, Auto Increment)

• ***user_id*:** Integer, References the student making payment (Foreign Key to user_id)

• ***course_id*:** Integer, References the course paid for (Foreign Key to course_id)

• ***amount*:** Decimal, Payment amoutn (Required)

• ***payment_status*:** Enum ('Pending', 'Completed', 'Failed'), Payment status (Required)

7. Feedback Table:

• ***feedback_id*:** Integer, Unique identifier for feedback (Primary Key, Auto Increment)

• ***user_id*:** Integer, References the student giving feedback (Foreign Key to user_id)

• ***course_id*:** Integer, References the course (Foreign Key to course_id)

• ***rating*:** Integer, Rating given (1-5 scale, Required)

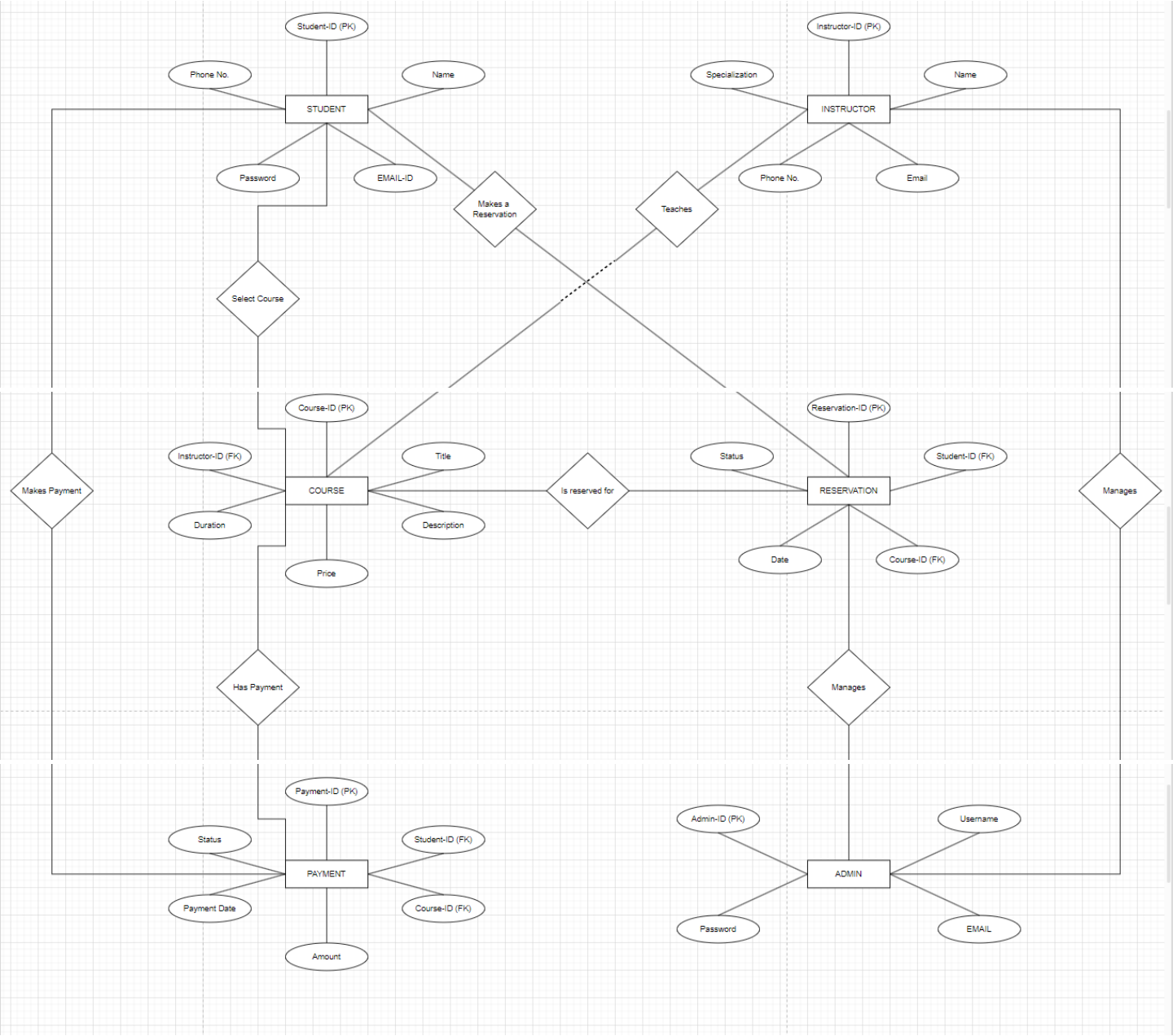
• ***comments*:** Text, Optional feedback comments

• ***feedback_date*:** DateTime, Timestamp of feedback submission (Default: Current Time)

EXPERIMENT-04

ENTITY RELATIONSHIP DIAGRAM

TOPIC: ONLINE COURSE RESERVATION SYSTEM



EXPERIMENT-05

SOFTWARE REQUIREMENTS SPECIFICATIONS

1) INTRODUCTION

1.1) PURPOSE: The purpose of this SRS document is to define the functional and non-functional requirements of on online course reservation system. This facilitates users to browse, select and reserve courses offered by an institution and the organization.

1.2) DOCUMENT CONVENTIONS:

- FR- Functional Requirements
- NFR- Non- Functional Requirements
- UC- Use cases

1.3) INTENDED AUDIENCE AND READING SUGGESTIONS: This section defines who the document is meant for and provide how different audience should navigate the document.

- Developers- For implementation and coding. And should focus on functional requirements and product requirements.
- Project Manager- Use for project planning and tracking process.
- End Users- Understand system capabilities and functions.
- Testers- Validation and Verifications.

1.4) PROJECT SCOPE: This system allows users to browse, select and reserve courses. This system also supports online payments and generate bills for administrators.

1.5) REFERENCES:

- IEEE SRS Standard (IEEE 830-1998)
- Academic policies of institutions.

2) OVERALL DESCRIPTION

2.1) PRODUCT PERSPECTIVE: This system is a web-based application that integrates users, admin, payment gateway and data managements systems.

2.2) PRODUCT FUNCTIONS:

- FR-1: User registration and Authentication.
- FR-2: Course Browsing and Searching.
- FR-3: Course Reservation.
- FR-4: Payment processing.
- FR-5: Notification system.
- FR-6: Admin Panel for course Management.

2.3) USER CHARACTERSTICS:

- USER: Reserve course and make payment.
- ADMINISTRATORS: Manage courses, enrollments and reports.

2.4) DESIGN AND IMPLEMENTATION CONSTRAINTS:

- BROWSER COMPATIBILITY: System must be compatible for the latest version of chrome, brave, edge, Firefox etc.
- SECURITY COMPLAINCES: The system must be updated to standards for data protection and securely payments.
- INTEGRATION LIMITATIONS: The system must be integrated with pre-approved third-party gateways and student databases.
- HOSTING REQUIREMENTS: The system must be deployed on a cloud service like google cloud or AWS for scalability.

3) SPECIFIC REQUIREMENTS

3.1) FUNCTIONAL REQUIREMENTS:

- FR-1: USER REGISTRATION AND AUTHENTICATION
  3.1.1) User can sign up using email and social login.
  3.1.2) Password reset functionality must be available.
- FR-2: COURSE BROWSING AND SEARCHING
  3.1.3) Users can filter by category, instructor and availability.
- FR-3: COURSE RESERVATION
  3.1.4) User can reserve available seats.
  3.1.5) Users receive a confirmation email.
- FR-4: PAYMENT PROCESSING
  3.1.6) Supports multiple payment methods.
  3.1.7) Generates invoices after successful payment.
- FR-5: NOTIFICATION SYSTEM
  3.1.8) Sends confirmation email and reminders.
  3.1.9) Notifies users about upcoming courses.
- FR-6: ADMIN PANEL
  3.1.10) Allows adding/ editing course.
  3.1.11) Generates reports on enrollments and revenue.

3.2) NON-FUNCTIONAL REQUIREMENTS:

- NFR-1: Scalability (System must support at least 5000 active users)
- NFR-2: Security (Implement OAuth 2.0 for Authentication)
- NFR-3: Pages should load within 3 seconds.

4) EXTERNAL INTERFACE REQUIREMENTS

4.1) USER INTERFACES

4.1.1) Responsive design for desktop, tablet and mobile devices.
4.1.2) Navigate with course search, filtering and reservation features.

4.2) HARDWARE INTERFACES

4.2.1) Compatible with standard desktop, laptops and mobile devices.
4.2.2) Server specifications required for hosting.

4.3) SOFTWARE INTERFACE

4.3.1) Integration with student management systems to fetch user details.
4.3.2) Connections with payment gateways.
4.3.3) APIs for sending emails and notifications.

4.4) COMMUNICATION INTERFACES

4.4.1) HTTPS for secure web communications.
4.4.2) REST APIs for data exchange between frontend and backend.
4.4.3) OAuth 2.0 Authentications for secure user login.