CS4400 Database Project
GTPort
Phase II

Team Members:

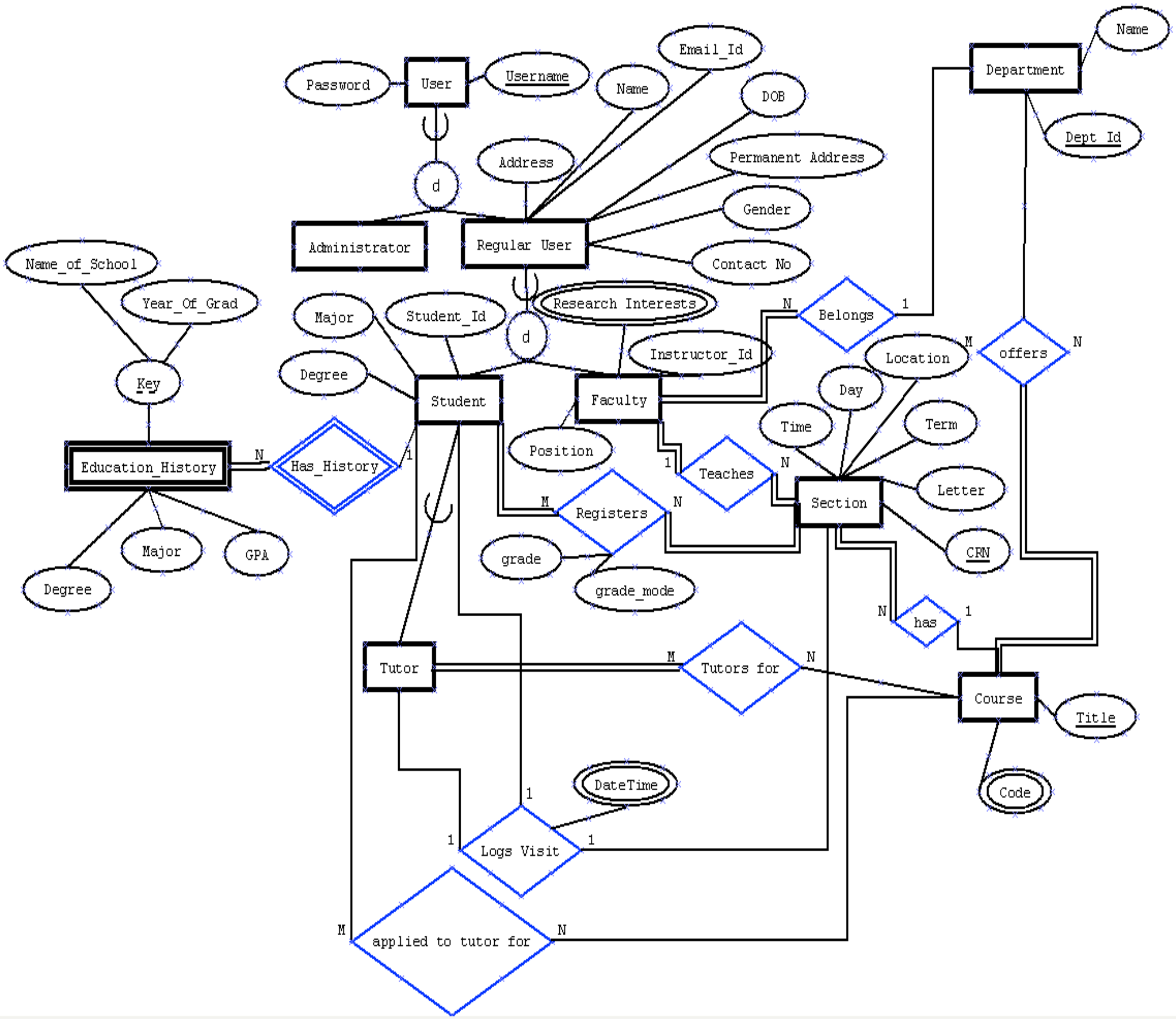
Lin Dong
ldong247
ldong247@gatech.edu
Section A

Nikhil Nandish
nnandish3
nnandish3@gatech.edu
Section B

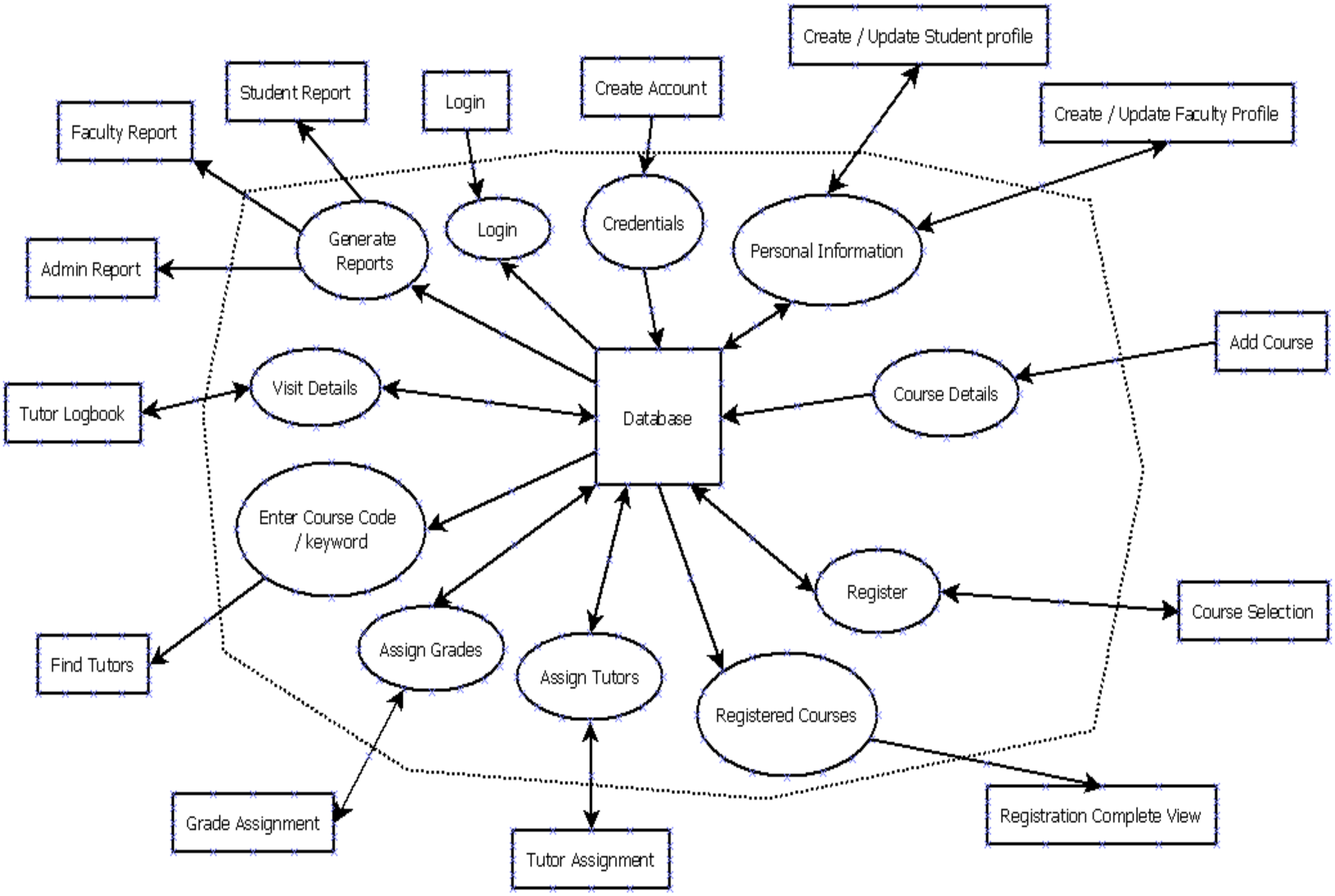
Stephen Gadd
sgadd3
sgadd3@gatech.edu
Section A

Linshu Wuliu
lwuliu3
lwuliu3@gatech.edu
Section C

1. EER Diagram



2. Information Flow Diagram


Create / Update Student profile
Create / Update Faculty Profile
Student Report
Login
Create Account
Faculty Report
Generate Reports
Login
Credentials
Personal Information
Admin Report
Add Course
Visit Details
Course Details
Database
Tutor Logbook
Enter Course Code / keyword
Register
Course Selection
Find Tutors
Assign Grades
Assign Tutors
Registered Courses
Registration Complete View
Grade Assignment
Tutor Assignment

3. Task Decomposition Diagram

4. Relational Schema Diagram

User

Username	Password

Administrator

Username

Regular_User

Username	Address	Name	Email_Id	DOB	Permanent_Address	Gender	Contact_No

Student

Username	Student_Id	Major	Degree

Education_History

Username	Year_Of_Grad	Name_Of_School	Major	Degree	GPA

Tutor

Username

Department

Dept_Id	Name

Faculty

Username	Instructor_Id	Position	Dept_Id

Research_Interests

Username	Interest

Course

Title

Code

Title	Code

Section

CRN	Term	Letter	Location	Day	Time	Instructor_Username	Title

Applied_To_Tutor_For

Username	Title

Tutors_For

Username	Title

Logs_Visit

Student	Tutor	CRN	DateTime

Registers

Username	CRN	Grade	Grade_Mode

Department_Offers_Course

Dept_Id	Title

5. Create Table statements

```
CREATE TABLE User
        (Username   VARCHAR(16)       NOT NULL,
         Password   VARCHAR(16)       NOT NULL,
PRIMARY KEY (Username));

CREATE TABLE Administrator
        (Username   VARCHAR(16)       NOT NULL,
PRIMARY KEY (Username)
FOREIGN KEY (Username) REFERENCES User(Username),
        ON DELTE CASCADE      ON UPDATE CASCADE);

CREATE TABLE Regular_User
        (Username             VARCHAR(16)       NOT NULL,
         Address              VARCHAR(30)       NOT NULL,
         Name                 VARCHAR(20)       NOT NULL,
         Email_Id             VARCHAR(30) NOT NULL,
        DOB                   DATE              NOT NULL,
         Permanent_Address          VARCHAR(30)       NOT NULL,
        Gender                CHAR(1)       NOT NULL,
        Contact_No            INT               NOT NULL,
PRIMARY KEY (Username),
UNIQUE (Email_Id),
FOREIGN KEY (Username) REFERENCES User(Username),
        ON DELETE CASCADE    ON UPDATE CASCADE);

CREATE TABLE Student
```

```
        (Username     VARCHAR(16)        NOT NULL,
         Student_Id   INT            NOT NULL AUTO_INCREMENT,
         Major        VARCHAR(33)        NOT NULL,
         Degree       VARCHAR(4)         NOT NULL,
PRIMARY KEY (Username),
UNIQUE (Student_Id),
FOREIGN KEY (Username) REFERENCES Regular_User(Username),
        ON DELETE CASCADE    ON UPDATE CASCADE);

CREATE TABLE Tutor
        (Username     VARCHAR(16)        NOT NULL,
PRIMARY KEY (Username),
FOREIGN KEY (Username) REFERENCES Student(Username),
        ON DELETE CASCADE    ON UPDATE CASCADE);

CREATE TABLE Education_History
        (Username             VARCHAR(16)        NOT NULL,
         Year_Of_Grad             INT            NOT NULL,
         Name_Of_School       VARCHAR(40)        NOT NULL,
         Major                VARCHAR(33)        NOT NULL,
         Degree               VARCHAR(4)         NOT NULL,
         GPA                  DEC(2,1)       NOT NULL,
PRIMARY KEY (Username, Year_Of_Grad, Name_Of_School),
FOREIGN KEY (Username) REFERENCES Student(Username),
        ON DELETE CASCADE    ON UPDATE CASCADE);

CREATE TABLE Department
        (Dept_Id      INT            NOT NULL AUTO_INCREMENT,
         Name         VARCHAR(33)        NOT NULL,
PRIMARY KEY (Dept_Id));

CREATE TABLE Faculty
        (Username             VARCHAR(16)        NOT NULL,
         Instructor_Id        INT            NOT NULL AUTO_INCREMENT,
        Position              VARCHAR(19)        NOT NULL,
         Dept_Id              INT            NOT NULL,
PRIMARY KEY (Username),
UNIQUE (Instructor_Id),
FOREIGN KEY (Username) REFERENCES Regular_User(Username),
        ON DELETE CASCADE    ON UPDATE CASCADE,
FOREIGN KEY (Dept_Id) REFERENCES Department(Dept_Id),
        ON DELETE CASCADE    ON UPDATE CASCADE);

CREATE TABLE Research_Interests
```

```
        (Username       VARCHAR(16)             NOT NULL,
         Interest       VARCHAR(20)             NOT NULL,
PRIMARY KEY (Username, Interest),
FOREIGN KEY (Username) REFERENCES Faculty(Username),
        ON DELETE CASCADE    ON UPDATE CASCADE);

CREATE TABLE Course
        (Title  VARCHAR(30)             NOT NULL,
PRIMARY KEY (Tiltle));

CREATE TABLE CCode
        (Title  VARCHAR(30)             NOT NULL,
         Code   VARCHAR(9)              NOT NULL,
PRIMARY KEY(Title, Code),
FOREIGN KEY (Title) REFERENCES Course(Title),
        ON DELETE CASCADE    ON UPDATE CASCADE);

CREATE TABLE Section
        (CRN                    INT             NOT NULL AUTO_INCREMENT,
         Letter                 CHAR(1)         NOT NULL,
         Term                   VARCHAR(6)              NOT NULL,
         Building               CHAR(4)         NOTL NULL,
         Location               VARCHAR(9)              NOT NULL,
        Time                    VARCHAR(19)             NOT NULL,
        Instructor_Username             VARCHAR(16)             NOT NULL,
        Title                   VARCHAR(30)             NOT NULL,
PRIMARY KEY (CRN),
FOREIGN KEY (Instructor_Username) REFERENCES Faculty(Username),
        ON DELETE CASCADE    ON UPDATE CASCADE);
FOREIGN KEY (Title) REFERENCES Course(Title),
        ON DELETE CASCADE    ON UPDATE CASCADE);

CREATE TABLE Applied_To_Tutor_For
        (Username       VARCHAR(16)             NOT NULL,
         Title          VARCHAR(30)             NOT NULL,
PRIMARY KEY (Username, Title),
FOREIGN KEY (Username) REFERENCES Student(Username),
        ON DELETE CASCADE    ON UPDATE CASCADE,
FOREIGN KEY (Title) REFERENCES Course(Title),
        ON DELETE CASCADE    ON UPDATE CASCADE);

CREATE TABLE Logs_Visit
        (Tutor          VARCHAR(16)             NOT NULL,
         Student        VARCHAR(16)             NOT NULL,
       CRN        INT         NOT NULL,
       DateTime   TIMESTAMP NOT NULL,
PRIMARY KEY (Student, Tutor, CRN, DateTime),
FOREIGN KEY (Student) REFERENCES Student(Username),
      ON DELETE CASCADE    ON UPDATE CASCADE,
FOREIGN KEY (Tutor) REFERENCES Tutor(Username),
      ON DELETE CASCADE    ON UPDATE CASCADE,
FOREIGN KEY (CRN) REFERENCES Section(CRN),
      ON DELETE CASCADE    ON UPDATE CASCADE);

CREATE TABLE Tutors_For
      (Username   VARCHAR(16)       NOT NULL,
       Title      VARCHAR(30)       NOT NULL,
PRIMARY KEY (Username, Title),
FOREIGN KEY (Username) REFERENCES Tutor(Username),
      ON DELETE CASCADE    ON UPDATE CASCADE,
FOREIGN KEY (Title) REFERENCES Course(Title),
      ON DELETE CASCADE    ON UPDATE CASCADE);

CREATE TABLE Registers
      (Username   VARCHAR(16)       NOT NULL,
       Grade            CHAR(1),
 Grade_Mode       VARCHAR(10)       NOT NULL,
 CRN        INT         NOT NULL,
PRIMARY KEY (Username, CRN),
FOREIGN KEY (Username) REFERENCES Student(Username),
      ON DELETE CASCADE    ON UPDATE CASCADE,
FOREIGN KEY (CRN) REFERENCES Section(CRN),
      ON DELETE CASCADE    ON UPDATE CASCADE);

CREATE TABLE Department_Offers_Course
      (Dept_Id    INT         NOT NULL,
       Title      VARCHAR(30)       NOT NULL,
PRIMARY KEY (Dept_Id, Title, Code),
FOREIGN KEY (Dept_Id) REFERENCES Department(Dept_Id),
      ON DELETE CASCADE    ON UPDATE CASCADE,
FOREIGN KEY (Title, Code) REFERENCES CCode(Title, Code),
      ON DELETE CASCADE    ON UPDATE CASCADE);
```

6. SQL statements for each task