UNG Automated Attendance System

Jordan Todd
Dr. Wei
Spring 2018
CIS 4950: IS Capstone
University of North Georgia

Abstract

This report provides the details from developing an automated attendance system for the University of North Georgia. The phases from system analysis and design, as well as the stages of project management are discussed in detail. By integrating a web interface, an application server, a database and a magnetic stripe card reader with the university's identification cards, an automated attendance system was developed. The system provides management functionalities for the university's instructors to maintain their courses attendance records. The system allows students to view attendance records collected in their courses. An instructor and a student are the recognized system users. The system is accessed by providing the correct URL in a web browser. The page loaded from correct URL input displays a log-in form. Valid credentials allow system access and the application server determines what type of user is accessing the system. A student user is presented with a webpage that enables course selections for viewing attendance records. An instructor user is presented with a webpage that initializes the management functions. Instructors select a course for which they are currently teaching, then they are prompted to select a task with the options as: collect, review, or update. Upon receiving the instructor's selections, the system redirects the instructor to the initialized webpage. The page loaded displays web controls and data objects relevant to the task selection.

I. Overview

1. **Introduction:** To fulfill its institutional goals, the University of North Georgia (UNG) provides a statement in their strategic plan declaring, "UNG will successfully leverage current and emerging technologies to support the quality of the student learning experience and operational effectiveness" (Engaging UNG). Currently, attendance is recorded manually by instructors writing roll-call results on paper to later input on D2L, inputting the attendance as they call roll, or collecting an in-class assignment from attending students. These recording methods take up too much time and are prone to error. A course with many students uses learning time to perform a simple, but necessary task. In the article titled, "A 21st-Century Attendance Policy" a teacher describes collecting attendance by using five minutes of every class for students to do an assignment. Instructors and students lose multiple full class meetings after totaling the five minutes used for each class in a semester. Many instructors include attendance as part of a student's final grade. Manually recording attendance jeopardizes the integrity of the collected data, which decreases the validity of a student's final grade. Researchers conducting a study on the errors of human data entry found, "visual checking resulted in 2958% more errors," than other methods studied (Barchard & Pace, 2011). Visual checking would be the method UNG instructors are using when submitting attendance records to D2L. The written roll-call results and the collected in-class assignments may get lost or damaged before being input on D2L, which would affect calculated grades. By implementing an automated attendance system, integrity is maintained for student attendance records, productivity increases, and efficiency increases throughout the institution.

2. **Problem Statement:** Ung manually records attendance, this method is highly capable of producing errors and reduces the amount of learning time.

3. **Project Summary:** This project developed an automated attendance system for the University of North Georgia. The purpose of the system is to provide error reducing automation functions for ensuring attendance data integrity and accelerating attendance management tasks. The system will allow instructors to perform restricted create, read, update, and delete (CRUD) operations for their courses attendance data. Students will swipe their issued identification cards in a card reading device and have their attendance recorded. An application server will execute the formatting process for swiped data and transmit the result to a database for storage. The application server will also process web requests and database query results for displaying data on the user interface.

4. **Project Objectives:** To meet the functional requirements the following objectives were identified and met:

   - Identification and selection of a low-cost magnetic stripe card reader
   - A scalable database was designed and developed for storing data objects
   - Data manipulation instructions were written for the database to process requests
   - An application was developed for data transmission between components
   - User-friendly web forms and reports were designed and developed
   - Attendance management functions were implemented in the system
   - The web application provides an interface, allowing users to access the system from any browser with web access

5. **Project Benefits:** The project will provide the following benefits:

   - Increase attendance record integrity
   - Increase student accountability
   - Decrease time required to manage attendance
   - Provide automation support
   - Validate student attendance grades
   - Support administrative decisions making

6. **Project Deliverables:** Upon accepting the project’s completion, the institution will receive the following integrated components:

   - Web Application
   - Application Server
   - Database
   - System Documentation

7. **Milestones:** For successfully meeting the projects objectives and delivering the system as proposed, the following dates were identified as major progress points:

   - 01/31/18 – Purchase Card Reader
   - 03/09/18 – Database Designed
   - 04/16/18 – Testing Completed
   - 02/18/18 – Working Prototype
   - 03/23/18 – Integration Complete

8. **Required Technology:** The following hardware and software technologies were required to develop the system:

   - Computer with network access
   - Web browser with internet access
   - .NET Framework
   - Microsoft SQL Server
   - Microsoft Visual Studio
   - Magnetic Stripe Card Reader

9. **Implementation Plan:** To meet the objectives the following components were integrated:

Component	Function	Source
Database	Performs CRUD Operations	Developed using T-SQL in SQL Server
Application Server	Processes Data Requests	Developed using .NET Framework and C#
Web Application	Provides Web Interface	Developed using .NET Framework and standard W3C markup languages

10. **Expected Outcome:** Successful implementation will provide UNG with an automated attendance system. The system will maintain the integrity of attendance records for the institution by providing accurate and consistent data collection, storage, retrieval, and modification functions. The automation will increase UNG's ability to provide a quality learning experience by decreasing the time used for collecting attendance. Students will further develop responsibility from being held accountable for their attendance grade and the automated system will validate the grade they receive. UNG administrators will be able to extract valuable information from the collected data, increasing their decision-making abilities, and furthering the institutions operational effectiveness.

II. Analysis

1. **Determining Requirements:** The systems requirements were initialized from studying attendance systems to identify their data structures and processing procedures. Understanding the organizational components is necessary for collecting the requirements of the proposed system. The institutions information needs, objectives as a business, key events, and data processing rules were significant factors considered during the analysis phase. The requirement collection method was traditionally approached by studying existing systems and the institutions business documents. Documentation was found by navigating the institutions website. A site search using the keyword "attendance" returned numerous results for identifying important organizational components.

Concrete examples provided direction and enabled the visualization of information and data in its practical use.

2. **<u>Structuring Requirements:</u>** After thorough investigation and observation of existing attendance systems, requirements were structured and organized as processes. To aid in the process modeling task, Microsoft Visio was used to create data flow diagrams. Process modeling enabled further understanding of the transmissions between a system and its environment for collecting, storing, and retrieving data. After process modeling, Microsoft Visio was also used to create conceptual and logical data models. Logic modeling was important for conveying the rules and interrelationships among the data objects. Logical data modeling also enabled a smooth transition to physical specifications for relational tables. *Diagrams are in the figure section at the end.*

III. Design

1. **<u>Front-End:</u>** The approach to designing the user interface was heavily influenced by the diagrams created in the analysis phase. Experience from previous projects in Microsoft's Visual Studio enabled a preliminary understanding of what web controls and interface objects were necessary to include in the web application. For conceptually visualizing necessary interface features a dialogue diagram was created. The diagram supported comprehension of the displays, sequences, iterations, and selections a system user is offered. A dialogue diagram box has three sections: the top section is unique to the display, the middle section is a simple descriptor name, and the bottom section is a reference to the displays that can be returned from the current display. The dialogue diagram is included in the figure section. To produce the final interface a prototyping methodology was implemented. Many iterations involving collecting info, building a prototype, assessing the usability, and adjusting appropriately, led to a finalized interface. The web application presentation design was influenced by the university's web toolbox webpage. The page offers a guide for designing and styling a webpage that aligns with the university's official branding and web design standards. Items from the guide that were implemented include the university brand color palette codes and the university brand logo images. Efforts were made to maintain the flex design and page element design, but technical skill and time constraints required a less restricted approach.

2. **<u>Back-End:</u>** The approach to implementing the back-end was traditional, involving integrating the database with the web application. An early prototype implemented a local SQL Server database but was discarded due to the portability limitations of a local database. An Azure server database replaced the local database and remained through the systems finalizations. In designing and creating the database schema the logical ERD was transitioned to a physical model of relations. The data dictionary in the figure section provides further details regarding the data types, primary and foreign keys, constraints, and field lengths.

   a. **<u>Data Access</u>:** Data access and transmission was achieved using a combination of ASP.NET's Entity Framework features and C#'s "SqlClient" namespace. The two of these techniques provided necessary methods for transmitting data between the web application and the database. The Entity Framework provided excellent data binding techniques for achieving effective and efficient data display. C#'s "SqlClient" methods were implemented for displaying data that didn't require a table. Both techniques were implemented to improve system response times and provide an optimized user interface experience.

IV. Implementation & Operation

1. **Coding:** Coding was simple to perform after the extensive planning, analysis, and design. Previous experience enabled quick back-end code implementation. Server-side logic, database creation and instructions for data definitions and manipulations were executed without issues. ASP.NET required research and experimentation for applying the necessary functions. Microsoft's .NET documentation, instructive websites, and YouTube tutorials were studied to overcome ASP.NET challenges.

2. **Testing:** Extensive desk checking, unit testing, and integration testing was performed on the system. However, of the various types of software application testing the most significant methods were not performed due to time constraints. Alpha and beta testing should be performed before production deployment occurs.

3. **Documentation:** System documentation was maintained throughout the development of the information system. External documentation is provided in the form of structured diagrams and models in the figure section. Source documentation can be downloaded from GitHub by searching "ung_autoattendance".

4. **Maintenance and Support:** The automation in the developed system requires little maintenance. To maintain the systems response times and ensure the systems data integrity routine maintenance should be performed. Error logs should be reviewed for vital updates and the database should be backed-up and restored at the end of a semester. The system has been designed to scale with the university's needs however extensive testing for scalability has not been performed and is recommended as a preliminary step to production deployment. If the university seeks additional support or contracts services from a 3rd party for any of the following reasons: making changes to the system for evolving its functionality, adding new features, improving performance, or making preventative risk avoidance changes; the system documentation should be reviewed for precautionary measures while working with this system.

V. Learning Experience

1. **Acquired Knowledge:** During the project's development extensive experience was gained in software application development and performing project management. The research performed to resolve development challenges greatly expanded programming knowledges and significantly progressed programming capabilities. Many hours were spent coding in Microsoft SQL Server and Microsoft Visual Studio. Comprehension of technologies and features within SQL Server and Visual Studio greatly expanded technical applications for using those program features. Examples of the technologies and processes are Microsoft's .NET Framework and the data access methods used in web application projects.

2. **Inaccurate Assumptions:** The project's initial work requirement assumptions were too generalized leading to challenges during development.

3. **Challenges:** The following challenges occurred while developing the system:

   - Learning Curve Delays
   - Ambiguous Requirements
   - Ill-Defined Scope

   The ambiguous requirements and ill-defined scope challenges were resolved by stalling further development to further define the project's details. The learning curve delays were resolved by committing extensive research time for the current task. Persistence led to greater knowledge which led to capabilities for solving the task-delaying problem.

References

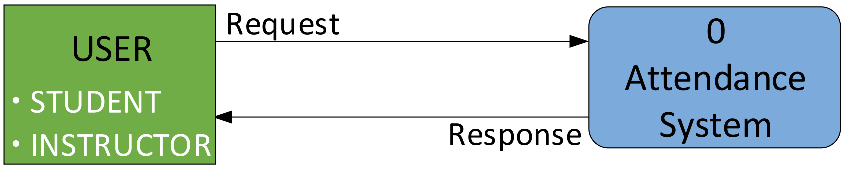
Jacobs, B. C. (2014, April). Engaging UNG [University of North Georgia Strategic Plan, 2014-2019].

LaFrance, M., & Corbett, S. J. (2014, July 14). A 21st-Century Attendance Policy. Retrieved January 29, 2018, from https://www.chronicle.com/article/A-21st-Century-Attendance/147693

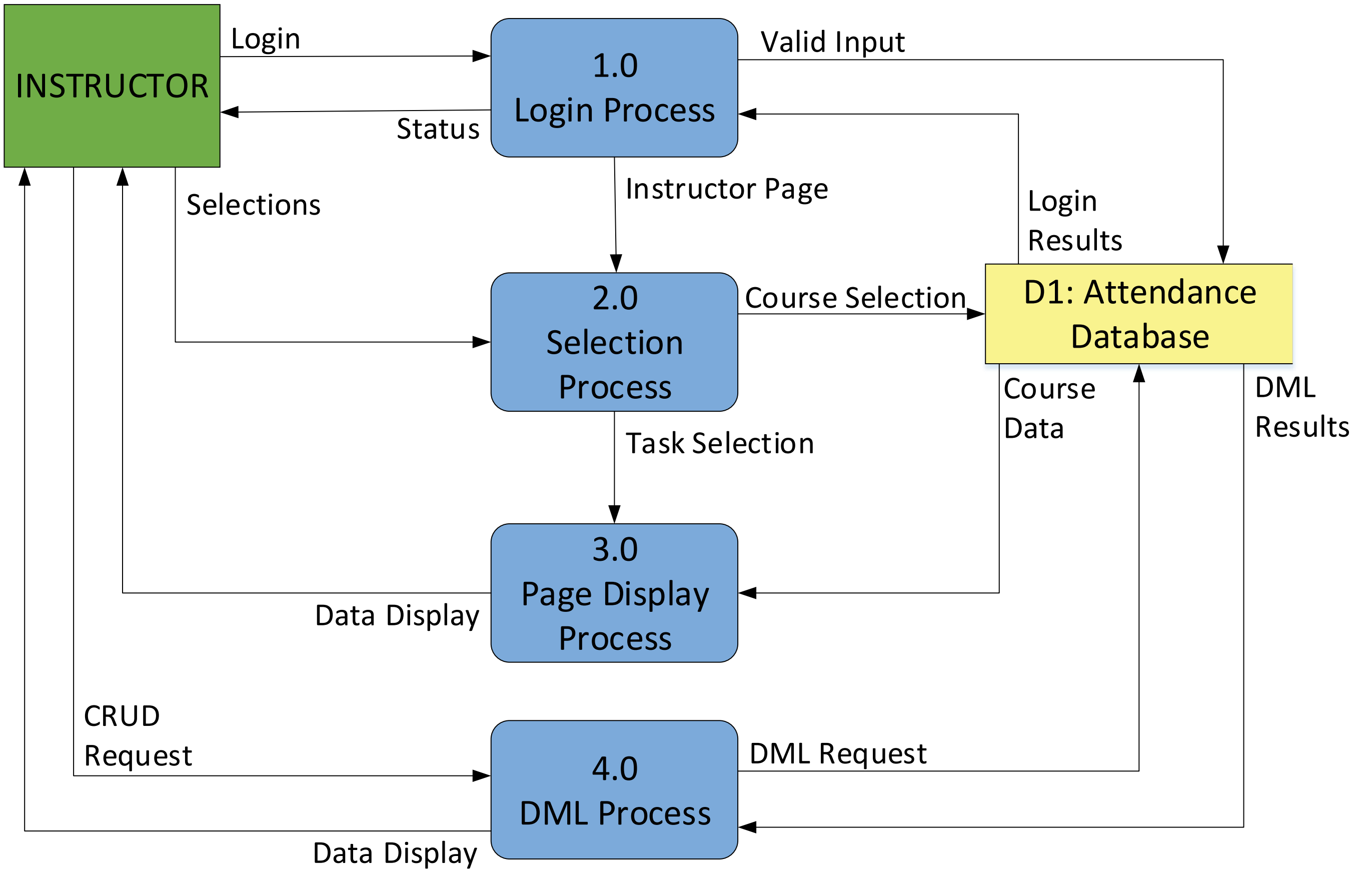
Barchard, K. A., & Pace, L. A. (2011). Preventing human error: The impact of data entry methods on data accuracy and statistical results. Computers in Human Behavior, 27(5), 1834-1839. doi:10.1016/j.chb.2011.04.004

Figures

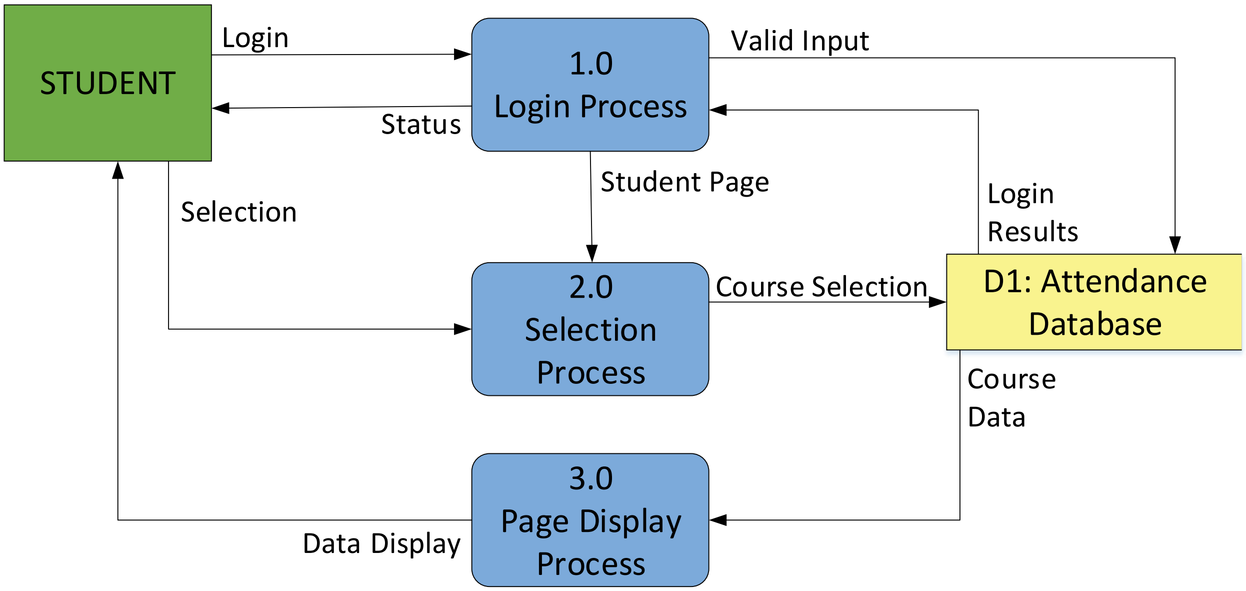
Context/Level-0 DFD



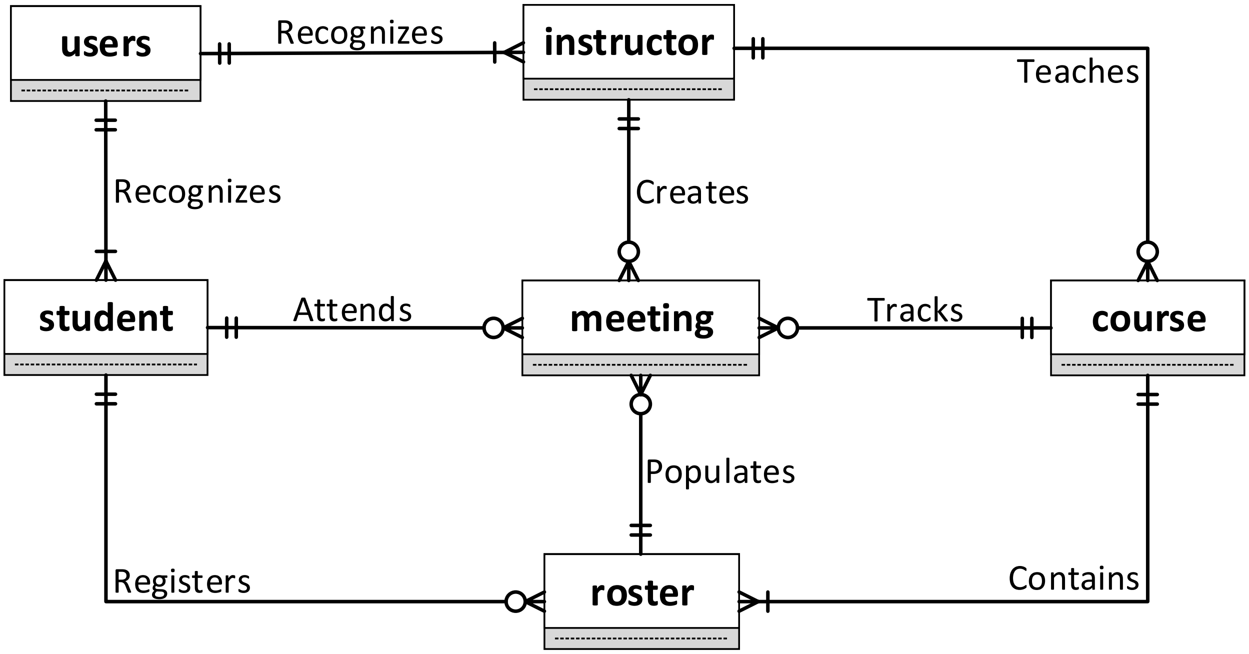
Level 1 DFD: Instructor User

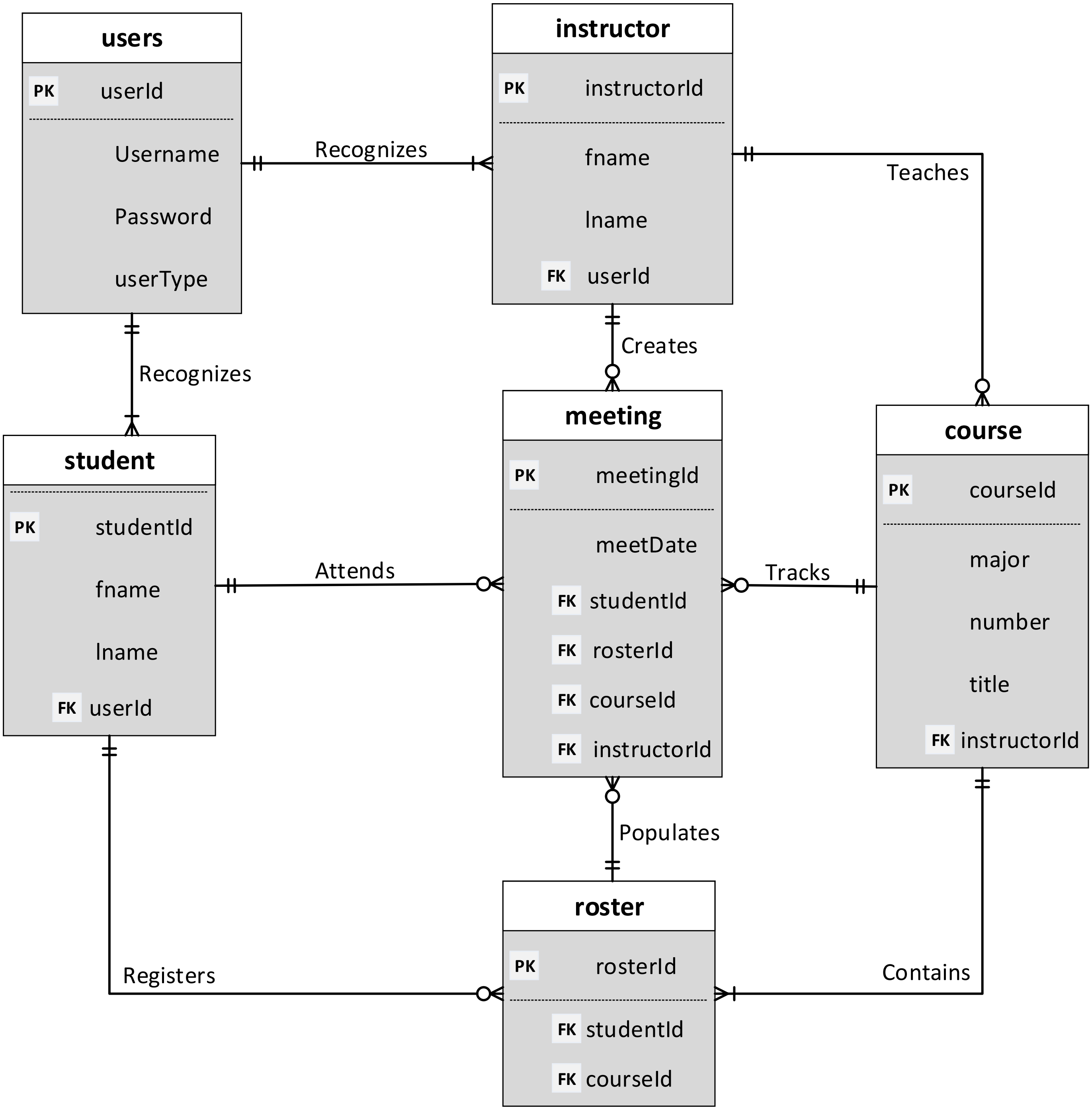


Level 1 DFD: Student User



ERD: Conceptual Model




users
PK userId
Username
Password
userType
instructor
PK instructorId
fname
lname
FK userId
Recognizes
Teaches
Creates
Recognizes
student
PK studentId
fname
lname
FK userId
meeting
PK meetingId
meetDate
FK studentId
FK rosterId
FK courseId
FK instructorId
course
PK courseId
major
number
title
FK instructorId
Attends
Tracks
Populates
roster
PK rosterId
FK studentId
FK courseId
Registers
Contains

Dialogue Diagram

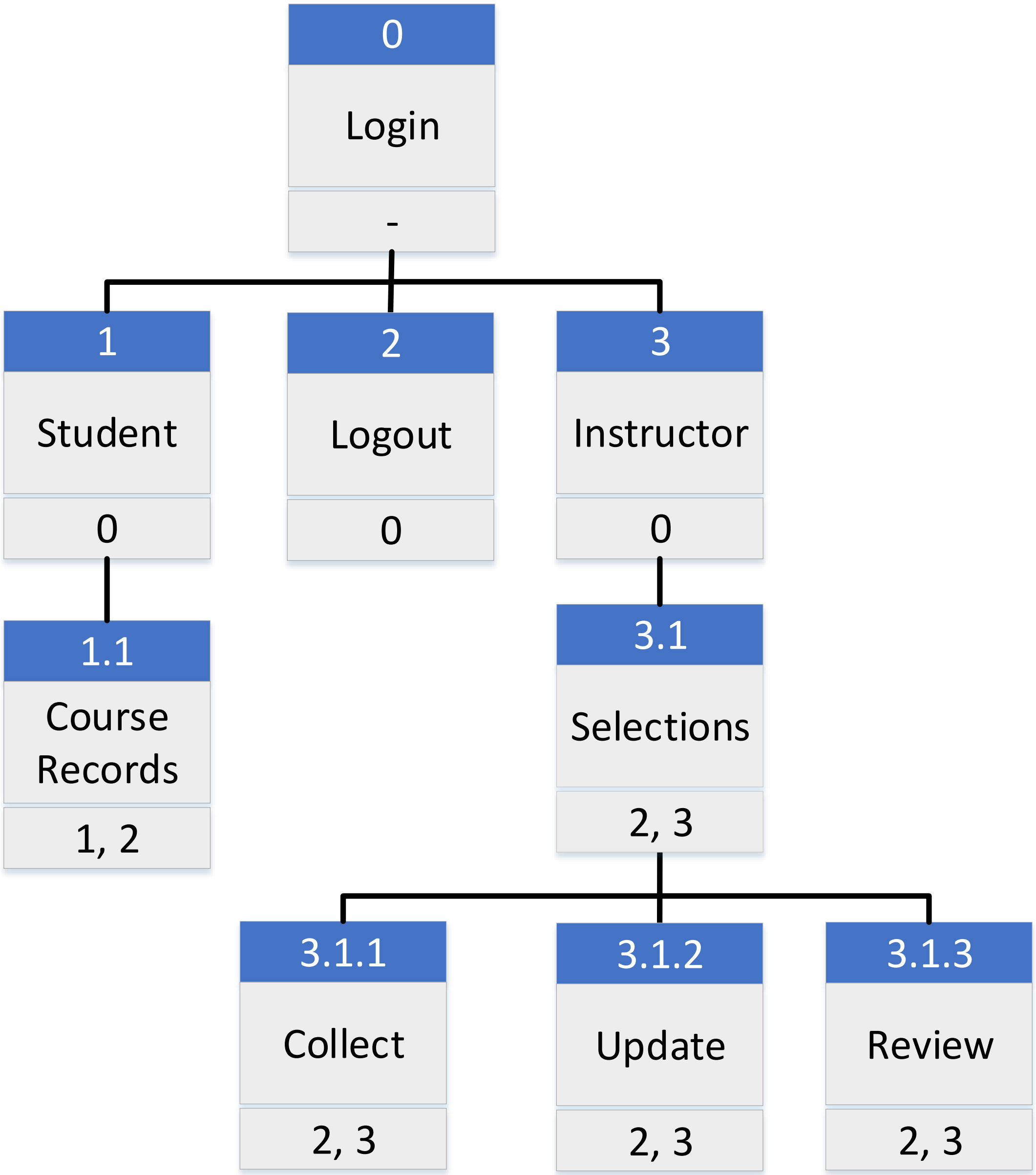


Table: USERS

Attribute Name	Data Type	Length	Key	Description	Format	Req?	Constraints	Example
userId	INT	-	PK	Uniquely identifies a users login	Identity	Y	-	22
Username	VARCHAR	20	-	Describes a users login name	Not Null	Y	-	John123
Password	VARCHAR	20	-	Describes a users login password	Not Null	Y	-	password321
userType	CHAR	1	-	Describes a users type	Not Null	Y	-	s

Table: INSTRUCTOR

Attribute Name	Data Type	Length	Key	Description	Format	Req?	Constraints	Example
instructorId	CHAR	9	PK	Uniquely identifies an instructor	Not Null	Y	-	900123456
fname	VARCHAR	20	-	Describes an instructors first name	Not Null	Y	-	Jane
lname	VARCHAR	20	-	Describes an instructors last name	Not Null	Y	-	Doe
userId	INT	-	FK	Uniquely id's an instructors login	Not Null	Y	users.userId	4

Table: STUDENT

Attribute Name	Data Type	Length	Key	Description	Format	Req?	Constraints	Example
studentId	CHAR	9	PK	Uniquely identifies a student	Not Null	Y	-	900756635
fname	VARCHAR	20	-	Describes a students first name	Not Null	Y	-	Jordan
lname	VARCHAR	20	-	Describes a students last name	Not Null	Y	-	Todd
userId	INT	-	FK	Uniquely identifies a students login	Not Null	Y	users.userId	1

Table: MEETING

Attribute Name	Data Type	Length	Key	Description	Format	Req?	Constraints	Example
meetingId	INT	-	PK	Uniquely identifies a course meeting	Identity	Y	-	45
meetDate	DATE	-	-	Describes the date for a meeting	Not Null	Y	-	01/01/18
studentId	CHAR	9	FK	Uniquely identifies a student	Not Null	Y	student.studentId	900123456
rosterId	INT	-	FK	Uniquely identifies a roster	Not Null	Y	roster.rosterId	2
courseId	CHAR	4	FK	Uniquely identifies a course	Not Null	Y	course.courseId	2011
instructorId	CHAR	9	FK	Uniquely identifes an instructor	Not Null	Y	instructor.instructorId	900123458

Table: COURSE

Attribute Name	Data Type	Length	Key	Description	Format	Req?	Constraints	Example
courseId	CHAR	4	PK	Uniquely identifies a course	Not Null	Y	-	2011
major	CHAR	4	-	Describes a courses major area	Not Null	Y	-	BUSA
number	CHAR	4	-	Describes a course number	Not Null	Y	-	3900
title	VARCHAR	30	-	Describes the courses title	Not Null	Y	-	Algebra II
instructorId	CHAR	9	FK	Uniquely identifies an instructor	Not Null	Y	instructor.instructorId	900876543

Table: ROSTER

Attribute Name	Data Type	Length	Key	Description	Format	Req?	Constraints	Example
rosterId	INT	-	PK	Uniquely identifies a roster	Identity	Y	-	32
studentId	CHAR	9	FK	Uniquely identifies a student	Not Null	Y	student.studentId	900765532
courseId	CHAR	9	FK	Uniquely identifies a course	Not Null	Y	course.courseId	2046