CS353 DATABASE SYSTEMS

2019-2020 SPRING

Project Proposal

for

Student Academic Information Registration System

TEAM 28 MEMBERS

SERVAN TANAMAN	*21201977*
MASTAN ABDULKHALİGLİ	*21403007*
DORUK ALTAN	*21401362*
OSMAN ORHAN UYSAL	*21602784*

1. Introduction

The project proposal describes clearly and fully the application system proposed by our team, and discusses why / how a database will be used as part of the system. Following this statement of the problem, the proposal defines the requirements and limitations of the application system. In addition, the report includes a conceptual database architecture using an E / R model.

E / R architecture involves strong and weak entities, binary/ternary relationships, aggregations, generalizations/specializations (e.g. IS-A relationship), restrictions on cardinality, key and descriptive attributes.

Moreover, we try to do a large number of entities, relationships, as long as our application permits. Our team tries to optimize code complexity in database design as allowed by our specifications. All entities and attributes are related to the project and there are no entities that are unnecessary.

This project plan was also published on a web page by the team members (in addition to submitting the printed proposal at its due date). The link and information related to the published pdf document can be found in the section related to it.

2. Project Description

Our system is basically a web-based application that offers a service for the students and instructors related to their registrations, information, grades etc. This

system is giving an opportunity to the Student Academic Information Registration System to effectively manage and maintain its data.

2.1 Problem Definition

In this project, we will be introducing a student academic information registration system. The project has students, instructors, teaching assistants, courses, and so on. To keep track of specific information, students should be able to sign in to the application. Students can do a variety of things like registering courses and editing account information like email and password. Students are also able to open a user interface which includes the information related to their individual student information, courses, attendances and also grades. Moreover, students may order an official transcript by providing a delivery address, choosing a method of delivery and making payment. Instructors should be able to submit grades and attendance for the students. Instructors may also authorize teaching assistants to perform specific tasks such as sending mail and submitting grades to students.

2.2 Importance of Database in Project

A database is typically designed so information can be easily stored and accessed. For any company or organization, a strong database is essential. This is because the database holds all related company documents such as payroll records, transaction records, compensation reports, etc. Therefore, because the Student Academic Information Registration System is a web-based application we need to store and use the information of the users of the system in order to ensure a good working system.

Student Academic Information Registration System cannot be thought without a database system behind it. This system consists of the information and data provided by the users which are students & instructors related to teaching assistants courses and so on. In order to view information for any user of the system, we need to store the data for the related part. Every functionality, objective, non-functional

requirements are live with the database system behind this Student Academic Information Registration System.

The database which we use behind this framework is the application core bone.

3. Requirements

3.1 Functional Requirements

3.1.1 Students

- Students are able to login to the system by using their ID number and password.
- Students can register for a selected course.
- Students can edit his/her own information such as email and password.
- Students can view his/her information.
- Students can view his/her courses.
- Students can view his/her grades.
- Students can view his/her attendance.
- Students can order a transcript that is official and given by the university to the specific address where the student wants to get a transcript.
- Students can also select a delivery method and payment terms.

3.1.2 Instructors

- Instructors are able to submit student grades.
- Instructors are able to submit student attendance.
- Instructors can also authorize teaching assistants to send an email to students.
- Instructors are able to authorize teaching assistants to submit grades.

3.1.3 Teaching Assistants

- TA can send emails to students related to instructors' authorization.
- TA can submit the grades of students.

3.1.4 Technical Support

- Technical support personnel can perform regular backups.
- Technical support personnel can recover disk space.
- Technical support personnel can do integrity checks to repair corrupted data.
- Technical support personnel can verify SQL backups.
- Technical support personnel can perform index defragmentation.

3.2 Non-Functional Requirements

3.2.1 Reliability

- Student Academic Information Registration System will be tested many times before it is released by the team members.
- All the team members will take a role during the testing phase because of the reliability.
- Team members will try to find out all possible scenarios which can bring a failure to the system and they will be tested many times.

3.2.2 Maintainability

- The architecture of the system will be easily maintainable.
- Any possible bugs or errors will be easy to detect.
- All possible bugs and errors can be fixed without any harm to the system.
- Student Academic Information Registration System also pays attention to updates and new features.

- Updates and new features if necessary will be extremely easy to adapt to the architecture of the system.
- Since the system can be used with mobile phones, Student Academic Information Registration System will be written with a native language in order to get a compatible application.
- New releases and versions will adapt to both IOS and Android applications with the help of the native language (FLUTTER).

3.2.3 Security

- KVKK laws will be considered and the system protects the data which are sensitive.
- All the data will be protected against malware attacks or access tries which are not authorized.
- The verification code will come to the user's approved email address and the user can get into the system if and only if he/she enters the correct verification code.
- The payment process during the transcription requirement will have a gateway which is PCI DSS compliant.

3.2.4 Usability

- The application has a simple entrance and the user can learn easily.
- The system is efficient because the user can reach his/her goal in few steps.
- The user can return to the interface after he/she clicked different interfaces and can start again.
- There will be a main tab panel in which the user can easily change the main interfaces.
- The system will be tested in order to calculate the usability on prototypes rather than on a finished product.

3.2.5 Availability

- The system will be available for 24 hours.

- Availability is the most important non-functional requirement because the users may use the system at any time.
- We can afford our application available 100 percent of the time.
- The system's different load scenarios are considered. For instance, during the add/drop week the system will experience a huge workload and the system will continue to serve with that overload.

3.3 Pseudo Requirements

- For the backend development of the website, we will use PHP.
- SQL is also crucial for the database and it will be used.
- Frontend development will be implemented with DART (Flutter) and also HTML and React will be used for the development.
- The system will be developed for the IOS, Android and also on the website.
- Genymotion will be used as an emulator for possible devices.
- Amazon DBS will be used and it will be the server of the system.
- Response time will not be more than 1 second.
- Queries in the database will be responded to within 2 seconds.
- The project has to be done before May 20 which is Demonstration day.
- Implementation will take place with 4 people.
- Course registration must take place at most in 3 seconds.

3.4 Additional Features

3.4.1 Course Swap Request

- Students can request a swap for a course. For example, if the student S1 has the Course C1 and S1 wants the course C2, in this situation another student S2 can swap the course C2, if and only if he or she has it already, with the course C1 if the S1 made a request for it.

3.4.2 Increasing Course Capacity

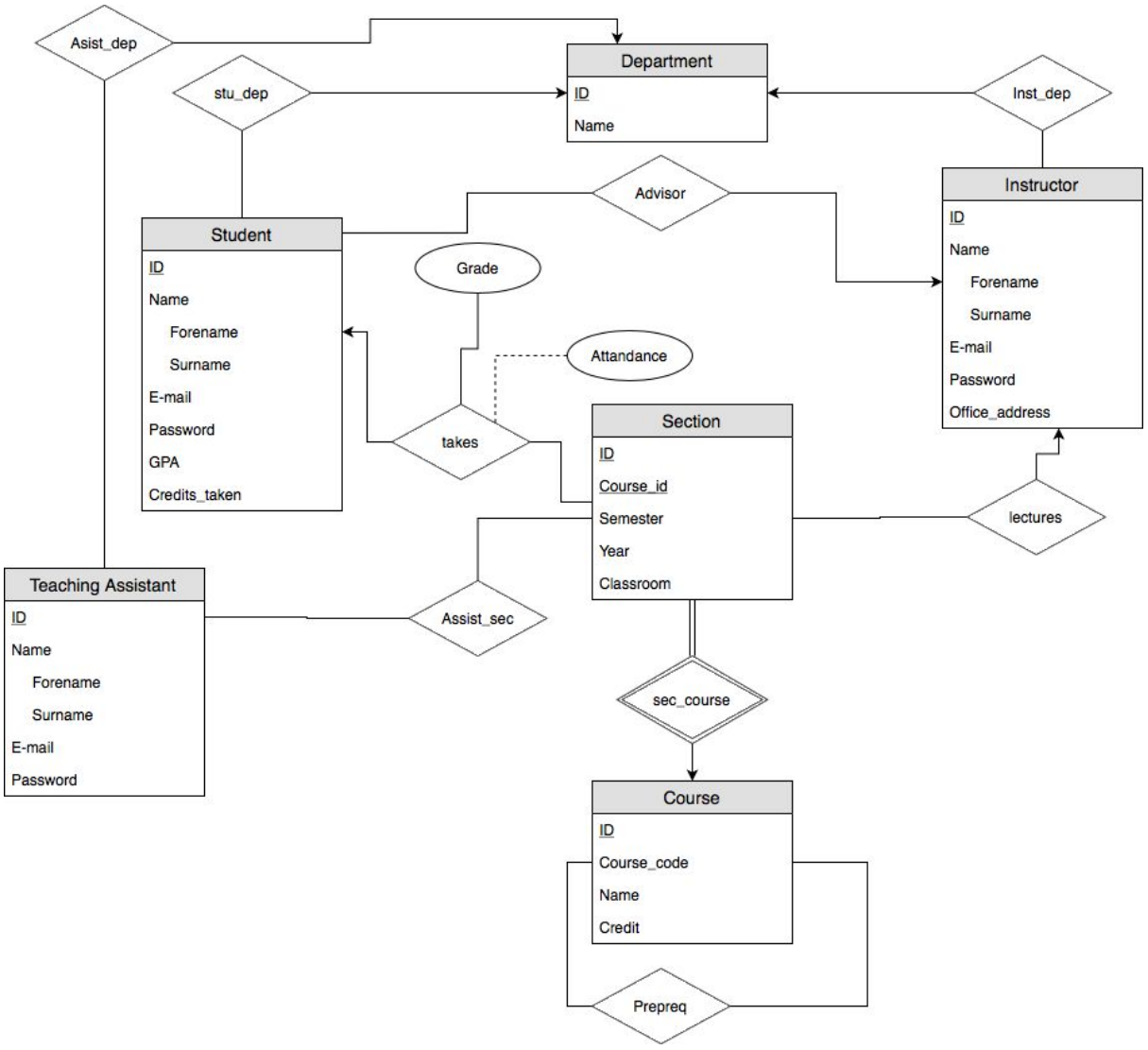
- Technical staff can open available quota for the course if necessary by the request of the instructor.
- Students can request available quota for the course and the instructor can decide to open a new quota or not.

4. Limitations

- Students can only view their own information.
- Students can only register to courses if they've satisfied the prerequisites for that course.
- Students can register up to a maximum of 8 courses.
- Students have 20 add/drop chances for day and night times.
- If a student's GPA is less than 2.00, he can register for at most 3 new courses.
- TAs can only send emails and submit grades with the instructor's permission
- TAs departments must match the instructors' departments.
- İnstructors can only view the information of students that are taking a course the instructor is currently giving.
- İnstructors cannot change sections.
- Students can make up to 10 swap requests for a semester.

5. Conceptual Design of the Database (E/R Model)

5.1 E&R Model



5.2 Entities

5.2.1 Department

- Holds a list of departments. Each department has a unique id and a name

5.2.2 Instructor

- Holds a list of instructors and their user information (UR). Fundamental user information in the system can be listed as unique id, full name, university email and password to login to the system. Instructors will also have office addresses.

5.2.3 Student

- Holds a list of students and their user information (UR). Each student will also have a cumulative GPA, and the total number of credits taken .

5.2.4 Teaching Assistant

- Holds a list of teaching assistants and their user information.

5.2.5 Section

- Holds a list of sections. Sections are bound to a course and it has the id of its course. İt also has year, semester(i.e. Summer 2020) and the classroom(like BZ-02) attributes.

5.2.6 Course

- Holds a list of courses. Each course has a unique id, a unique course code (like CS353), a full course name, and the number of credits the course has.

5.3 Relations

5.3.1 Asist_dep

- A table for a list of assistants in each department.

5.3.2 stu_dep

- A table for a list of students in each department.

5.3.3 Advisor

- A table for a list of instructors in each department.

5.3.4 takes

- A table holding which student took which section of a course. It also stores the final letter grade that that student got from that course as well as attendance ratio.

5.3.5 lectures

- A table holding which instructor lectures which sections.

5.3.6 Assist_sec

- A table for a list of assistants in a section as well as a list of sections an assistant has.

5.3.7 sec_course

- A table for a list of sections of any course.

5.3.8 Prereq

- A table holding a list of courses that another needs it as a prerequisite.

6. Conclusion

Our project is a web-based student academic information registration system similar to STARS that offers certain services to students, TAs and instructors. These services include registration, transcript ordering, viewing and submitting of grades. This report aims to provide the necessary details about all the important aspects of the design and implementation processes. Since the main purpose of the application is to store, retrieve and manipulate data, the DBMS(Database Management System)

behind the application is the heart of the system. Such systems deal with very large amounts of data which make efficiency a top priority both speed-wise and memory-wise . The previous sections illustrate the decisions and constraints of the design to satisfy the functional requirements for the users while maintaining the non-functional requirements such as reliability and maintainability.

The main goal of this proposal is to give a general understanding of how our system will work as a whole, show the roles of every major component, and give our reasons in the decision-making process as all these factors will influence the implementation process. To that end, the E/R diagram above graphically represents the structure of our system in its entirety. For more specific descriptions; user capabilities, necessary functions and limitations all have sections that show their important details individually and technical details like what programming language is used on presentation, application, and data tiers are given in the Pseudo Requirements section.

7. Website Information

https://github.com/MastanAbdulkhaligli/CS353-Project-Proposal