



MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

---

SMART METU CAMPUS

CENG 350

---

AYŞENUR BÜLBÜL  
2171403

ZEYNEP ERDOĞAN  
2171577

# Table Of Contents

1	Introduction . . . . .	5
1.1	Purpose of the System . . . . .	5
1.2	Scope . . . . .	5
1.3	System Overview . . . . .	6
1.3.1	System Perspective . . . . .	6
1.3.1.1	System Interfaces . . . . .	6
1.3.1.2	User Interfaces . . . . .	6
1.3.1.3	Hardware Interfaces . . . . .	7
1.3.1.4	Software Interfaces . . . . .	7
1.3.1.5	Communications Interfaces . . . . .	7
1.3.1.6	Memory . . . . .	8
1.3.1.7	Operations . . . . .	8
1.3.2	System Functions . . . . .	9
1.3.3	User Characteristics . . . . .	10
1.3.4	Limitations . . . . .	11
1.4	Definitions . . . . .	12
2	References . . . . .	13
3	Specific Requirements . . . . .	14
3.1	External Interfaces . . . . .	14
3.2	Functions . . . . .	17
3.3	Usability Requirements . . . . .	19
3.4	Performance requirements . . . . .	20

3.5	Logical database requirements . . . . .	20
3.6	Design constraints . . . . .	21
3.7	Software system attributes . . . . .	21

# List of Figures

1	Context Diagram . . . . .	9
2	External Interface Diagram . . . . .	14
3	Use Case Diagram . . . . .	17
4	Sequence Diagram for "Reporting Heating Problem" . . . . .	19
5	Logical Database Requirement Class Diagram . . . . .	20

# List of Tables

1	Functionalities of System . . . . .	10
2	Use case description for reporting a heating problem . . . . .	18
3	Use case description for reserving a spot . . . . .	18

# 1 Introduction

As the technology getting more advanced every single day, people's usage of it in their daily lives gets increases. One of the main usages of technology is making our living spaces 'smart'. Which makes it easier for people to remotely control and manage systems such as heating, security, lighting and electronic devices. This project is focused on making our university, Middle East Technical University smart.

## 1.1 Purpose of the System

The main purpose of the system is providing users to ease his/her tasks. For instance, Instructors can easily report a problem in a class to the related worker or students can easily book spots in library or computer labs or workers can easily check reported problems and mark them as solved.

## 1.2 Scope

This system is in web application and mobile application form. There are 4 users groups. First one is instructors. Instructors should be able to solve three main tasks:

- Taking attendance
- Reporting problems related to classrooms and user's account
- remotely controlling conditions of the class such as heating, lighting, projection etc.

Second group is students. Students should be able to solve four main tasks:

- Checking and reserving empty spots in library and computer labs
- Reporting problems related to user's account, computer, Wi-Fi etc.,
- Checking available books
- Participating attendance.

Third user group is workers. Workers should be able to solve two main tasks:

- Reporting user's account problems
- Checking and marking as solved all the problems that are reported

The last user group is admin. Admin should be able to solve two main tasks:

- checking and marking as solved account problems
- blocking the users/workers if they violate the rules

## 1.3 System Overview

### 1.3.1 System Perspective

System has 4 main parts: web application/mobile application, scanners, server and database. Web Application/mobile application is where users interact with. Servers manipulate the data, store it in databases and get it back from the database or update the data when asked.

#### 1.3.1.1 System Interfaces

The system has a system interface which is the web server. Users do not interact with the servers directly. After user-interface sends data to database, it first interact with servers. Servers manipulate the data, store it in databases and get it back from the database or update the data when asked. For instance, if a book is available, server finds that book from the database and send it's location back to the user-interface if it is available. Another example would be user checking empty spaces in the library or computer labs but this time servers constantly updates the empty spots.

#### 1.3.1.2 User Interfaces

Web application/mobile application has 2 pages. Login/register page and user's profile page.

**Register and Login Interface:** Login and register page is the same for every different user group but once users login, a different page comes up depending on their user group and their personal information.

**Instructor-User Interface:** The user-interface for the instructors has 2 main components. First one is where instructor can see and confirm attendance and the second one is where instructor can report problems. Problems are shown as dropdown list.

**Student-User Interface:** The user-interface for the students has 4 main components. First one is where they can see and reserve the empty spots in library. The second one is where they can see and reserve the empty spots in computer labs. The third one is where they can search a available book in library and the last one is where they can report problems. Empty spots and problems are shown as dropbox list. The last one is where they can see their attendance information.

**Worker-User Interface:** The user-interface for the workers has one main component where they can see the problems as list that are reported also they can mark the box next to problem if solved.

**Admin Interface:** admin interface has 3 main components. First where they can see reported problems related to users account and second where they can see all the action in the whole system. Last, where they can see all the errors server send.

#### 1.3.1.3 Hardware Interfaces

**Scanners:** Scanners are for students to participate attendance. They should be able to scan their student cards to the scanner which is located in every desk in the classes. Scanner then sends this data to related server.

#### 1.3.1.4 Software Interfaces

**Database:** Database contains the report records, the user account records, the spots records in library and computer labs, book records and (for attendance) course-student records. Used Database is MongoDB

**Operating System:** Operating system would be server distributions of Linux or Windows.

#### 1.3.1.5 Communications Interfaces

Server and the database use to communicate a built in module MongoDB provides which is called mongoose. Server and the user interface communicate using HTTP Modules.



#### **1.3.1.6 Memory**

Memory can be a problem for the system since the system will keep record of attendance information so system should have enough memory to keep all the users attendance information and to sustain the system.

#### **1.3.1.7 Operations**

The operations provided by the system can be divided into:

##### **Student Operations:**

- Register
- Login
- Report computer problems/ account problems
- Check empty spots in library/computer labs
- Reserve a spot in library/computer labs
- Check available books
- Scan ID card to scanner when in class

##### **Instructor Operations:**

- Register
- Login
- Report heating/ projector/Wi-Fi/account problems
- Confirm Attendance

##### **Worker Operations:**

- Register
- Login
- Report account problems

- Receive heating/ projector/Wi-Fi problems and marked the received problems as solved

#### Admin Operations:

- Block an user
- Receive account problems and marked the received problems as solved
- Have an access to all the other user-interfaces

#### System Operations:

- Get data from user interface and save them in the database
- Get requested data from the database send it to user interface.
- Report any problem on the system to admin interface.
- Gets attendance information from the scanners and inform the instructors user interface and stores it in the database if instructors confirms.

#### 1.3.2 System Functions

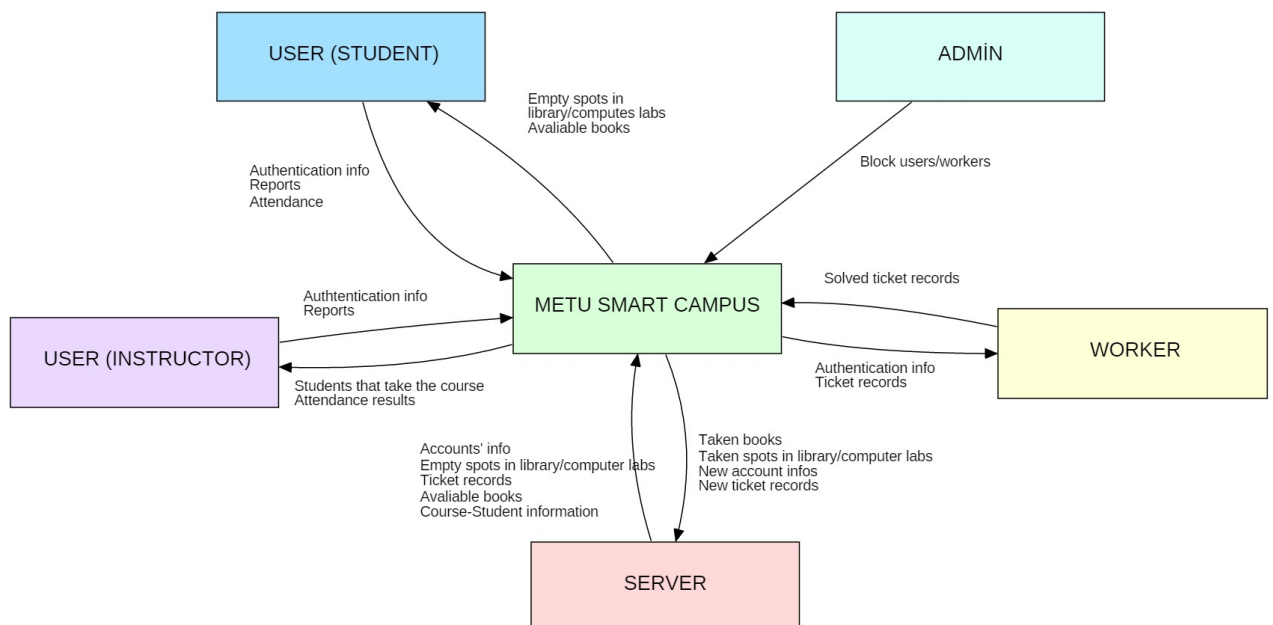


Figure 1: Context Diagram

No	Functionality	Description
1	Login	Let the user/worker login the smart campus system with ID number and password.
2	Logout	Let the user/worker logout from the smart campus system.
3	Register	Let new students, instructors or workers register to the smart campus system with their ID number to be able to use the system.
4	Block a user/worker	Let admin to block a user/worker if it is necessary.
5	Report problems with account	Let users/workers report the problems with their accounts to the system admin.
6	Report projector problems	Let instructors report problems with the projector in classrooms to the related workers.
7	Report heating problem	Let instructors report problems with heating in classrooms to the related workers.
8	Report Wi-Fi problems	Let instructors report the problems with Wi-Fi to the related workers.
9	Mark problems as solved	Let workers mark the problems as solved when he/she solves the problem.
10	Attendance	Let students scan their card to the screen and participate the attendance. Let instructors see the attendance result of the related class. Let instructor confirms the attendance
11	Check empty spots in library(Reserve a spot)	Let students check the availability of spots in library. If there is an empty spot let students reserve it for a short time.
12	Check available computers in campus(Reserve a spot)	Let students check the availability of spots in computer labs. If there is an empty spot let students reserve it for a short time.
13	Check available books	Let students search the book he/she is interested and check its availability.
14	Report computer problems	Let students report the problems with computers to the related worker or admin.

Table 1: Functionalities of System

### 1.3.3 User Characteristics

Target users of smart METU campus can be categorized into four types: instructors, students, workers and admins.

Instructors, students and workers need to have basic computer skills since their only task is using the interface. On the other hand, admins have to be expert in their fields since they manage the whole system.

#### 1.3.4 Limitations

- **Regularity policies:** The privacy of the users should not be published to community and must be protected at all costs.
- **Hardware Limitations:** Hardware should be fast enough to update the empty spots in real time.
- **Interfaces to other applications:** There is no other application connected to our smart METU campus system.
- **Parallel operation:** Users should be able to do their task simultaneously and don't get an error such as an empty spot can be reserved by only one student.
- **Audit functions and Control functions:** Accounts are controlled by the system admins to minimize security breach and system failure.
- **Higher-order language requirements:** System should be written in multi-program object-oriented programming languages. Since the project is considered to be used for a long time the technologies that are used should be updatable and used if there has to be a change in the system.
- **Signal handshake protocols:** Our website uses only HTTP protocol.
- **Quality requirements:** The system should do the backup regularly since the data of users is everything in our system.
- **Criticality of the application:** Since there can be emergencies in classrooms which prevents instructors to continue the lecture, admins should keep the system up all the time.
- **Safety and security considerations:** The security of accounts should be protected by the system admins. Such as passwords should be hashed while storing in database.
- **Physical/mental considerations:** System can be upgraded in the other versions for easing the usage of the students with disability.

## 1.4 Definitions

- **Student user:** Student at METU who has an account on the system.
- **Instructor User:** Instructor at METU who has an account on the system.
- **Worker:** Expert at specific field who is working at METU and has an account on the system.
- **Admin:** Expert who manages the entire system also the cyber security.
- **User Interface:** Web application or mobile application that is part of the system.
- **Database:** A database which can store the records of accounts and reports easily using MongoDB.
- **Scanner:** A sensor system which scans the ID card of the users and send the related data to the smart METU system.
- **HTTP protocol:** HTTP is a protocol which allows the fetching of resources, such as HTML documents.

## 2 References

**This document is written with respect to IEEE 29148-2011 standard:**

IEEE. (2011, December 1). 29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes –Requirements engineering. Retrieved from <http://ieeexplore.ieee.org/document/6146379/> on March 12, 2018. doi: 10.1109/IEEESTD.2011.6146379

### 3 Specific Requirements

#### 3.1 External Interfaces

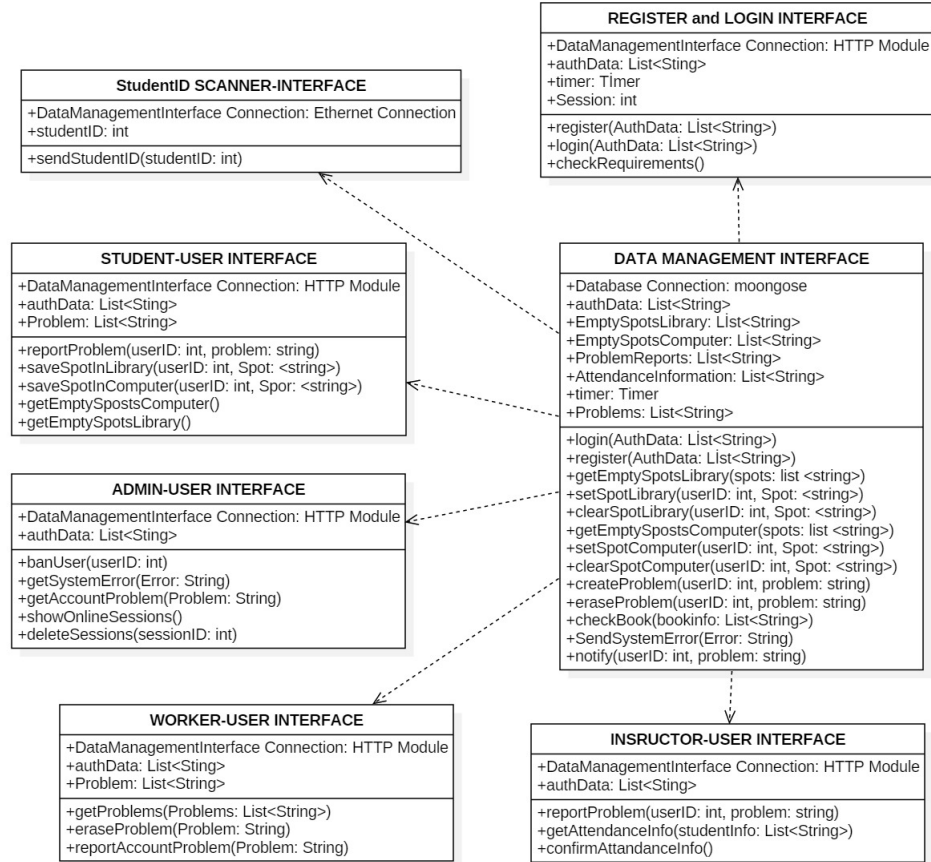


Figure 2: External Interface Diagram

- **Register & Login Interface**

This interface allows the new users to register to the system and the already registered users to login to system. Users fill the required information (ID number and password) and information sent to database to be validated. Wrong and missing submissions cause an error message. Refreshing the page clears all the not submitted information. Inputs are text inputs and button for sending information.

- **Student-User Interface**

There are 3 main functionalities that can be done via student interface. This interface allows students check empty spots in library and computer labs. Busy spots will be red and empty spots will be green. Students can click on empty spots and click the "Reserve" button to reserve the place for a short time. Also students can check available books via searching them in the search bar which has autocomplete feature. Students can report the computer problems via a page which has a form that is going to be filled with information of computers and problems.

- **Instructor-User Interface**

There are 2 main functionalities that can be done via instructor interface. This interface allows instructor report problems in a classroom via a page which has a form that is going to be filled with information of problems. Also there is an attendance interface where instructor can see the students that are present in the class. By clicking on them instructors can see the information about students.

- **Worker-User Interface**

This interface allows workers to see the reports as a list. By clicking the reports workers can see the information about the problem. Also there shall be a button "Solved". By clicking on it workers mark the problem as solved and this process deletes the report from the workers interface.

- **Student ID-Scanner Interface**

This interface scans the ID card of students. It is just like the system that we are using in labs doors but smaller.



- **Admin-User Interface**

This interface allows admins to see reported account problems. Admins can see the more information about them by clicking on them. Also admins can see all the accounts and by selecting "Block the user" admins can ban accounts.

- **Data Management Interface**

This interface shall categorize every entry such as reports, reservation information and send them to the related databases. This interface ,also , is responsible of admins activities such as login.

### 3.2 Functions



Figure 3: Use Case Diagram

<b>Use Case ID</b>	<b>1</b>
Use Case Name	Report heating problems
Actor	User(Instructors), Workers
Description	In case of a heating problem in a classroom instructors can report it to workers.
Pre-conditions	1. Instructors and workers should log into the system. 2. Instructors should navigate to heating problems section.
Post-conditions	1. Ticket is created according to specifications. 2. Worker is informed by the system.
Normal Flow	1. Instructor logs in the system. 2. Instructor creates a ticket, specifies; - The problem(heating/Wi-Fi etc.) - Building and classroom then sends the report. 3. The worker is informed by the system.
Exceptions	The instructor cannot create a ticket if the problem is already reported and not solved.

Table 2: Use case description for reporting a heating problem

<b>Use Case ID</b>	<b>2</b>
Use Case Name	Reserve a spot in library
Actor	User(Students)
Description	If there is an empty spot in library, student can reserve this spot for 5 minutes until he/she gets there.
Pre-conditions	1. Students should log into the system. 2. There has to be an empty spot in library.
Post-conditions	1. Spot is reserved for 5 minutes. 2. If the student doesn't get there on time spot will be seen as empty again.
Normal Flow	1. Student logs in the system. 2. There is an empty spot that can be reserved. 3. Student reserves the spot and the spot is displayed as busy for 5 minutes. 4. Student gets there and takes the spot.
Alternative Flow	4. Student doesn't take the spot on time, spot is displayed as empty again and the student has to wait for an hour to reserve a spot.
Exceptions	If there is no empty spot so student cannot reserve a spot.

Table 3: Use case description for reserving a spot

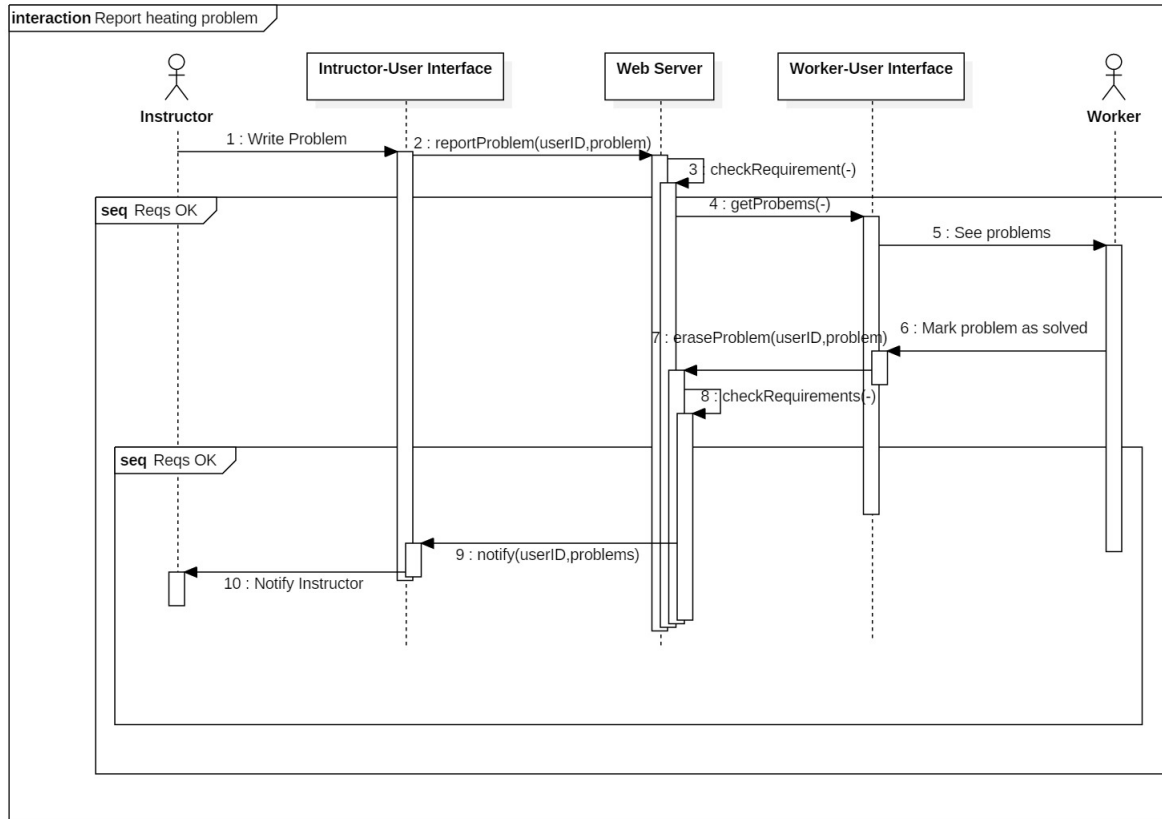


Figure 4: Sequence Diagram for "Reporting Heating Problem"

### 3.3 Usability Requirements

- For students, checking and reserving a spot application, page should be refreshed at regular intervals so the system can be effective.
- For students, while searching for a book, search bar should autocomplete and give suggestions to ease the process.
- For students or instructor, when they want to report a problem and that problem is already reported they should easily see the existing report when they choose the specific computer/classroom etc.
- After the attendance process the instructor should be able to see the students in the class and the information about them in the application. The view which shows who is

sitting where and the photo of the student should be there.

### 3.4 Performance requirements

- The system should be available to all students/instructors/workers simultaneously.
- Empty spots in library and computer labs should be updated in per minute.
- Attendance results should be reachable in a minute.
- All the reports should be sent to the database and can be viewed in workers' interface in at most 30 seconds.

### 3.5 Logical database requirements

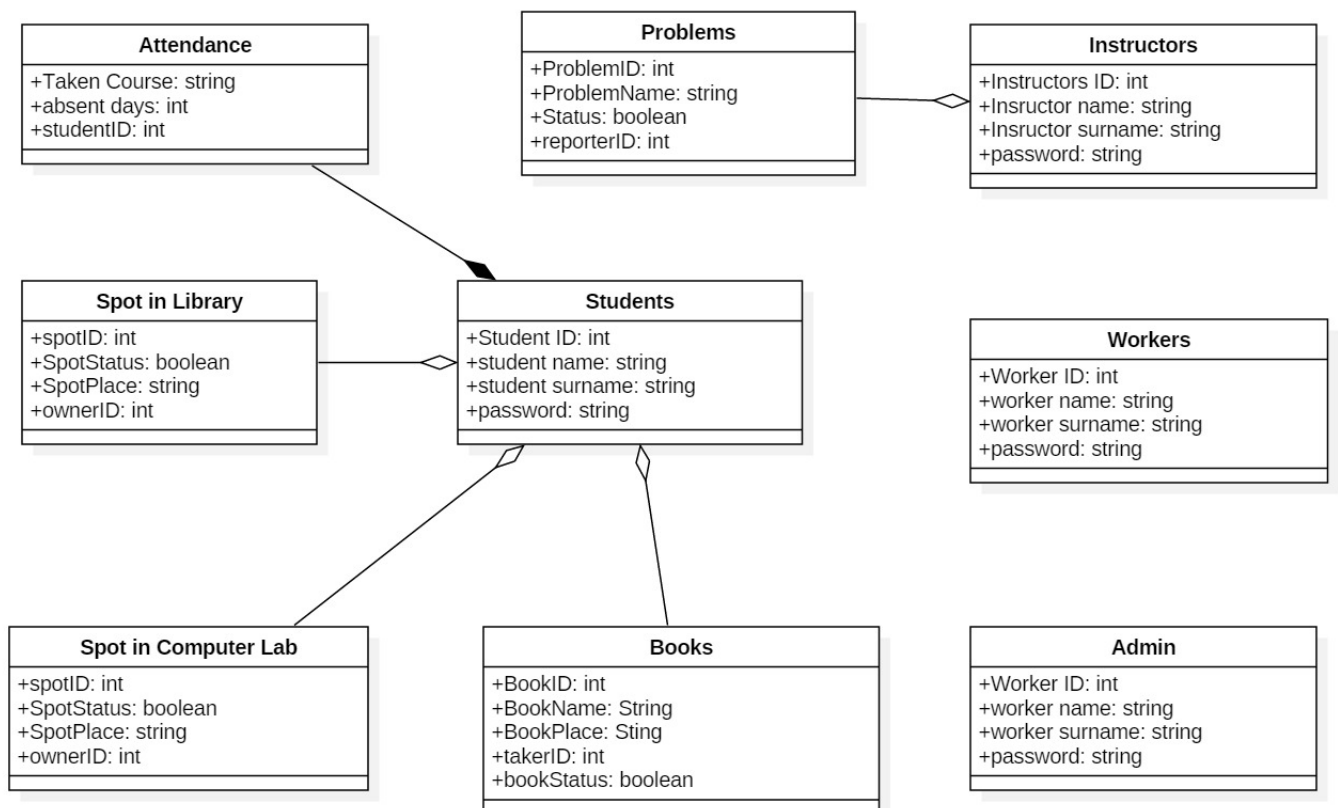


Figure 5: Logical Database Requirement Class Diagram

- Only admin can access to all of the tables.
- Only admin can add new admin .
- Only admin can erase a users account.
- Admin can erase a entry from problems entry (usually account problem.
- Student user can access to her/his information, books table, spot in computer lab/library table also student can add new entry to problems table (user account problem).
- Worker user can access to her/his information, problems table and can erase a problem.
- Instructor user can add new entry to problems table.
- Student can access to attendance table but cannot chance it.
- User can change status of a book
- User can change status of a spot in library
- User can change status of a spot in computer lab

### **3.6 Design constraints**

- All the reservation information about students shall be stored for legal purposes.
- All the report information about the person who did the report shall be stored for legal purposes.
- The system shall be designed in accordance with law of privacy.

### **3.7 Software system attributes**

#### **1. Reliability:**

Report messages should be checked frequently.

#### **2. Availability:**

System shall be available to users when they have internet connection. Maintenance

operations shall be done once a month between 3 AM and 6 AM.

**3. Security:**

Since smart METU campus system has important information of students/instructors/workers (for example TC id number, phone number, METU id number etc. can be stored, needed information can be changed by time) they shall be protected and not be accessible except system admins.

**4. Maintainability:**

Documentation of every feature should be easily understandable so that new admins/ IT workers integrate the system quickly. Developers are responsible of the documentation of feature they create and adaptation of the system's new needs.

**5. Portability:**

System should be available in both mobile devices and web browsers. Users should install the app in their mobile phones to be able to use the system. In a web browser users just need to login to use the system.