MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING

SMART METU CAMPUS

CENG 350

AYŞENUR BÜLBÜL
2171403

ZEYNEP ERDOĞAN
2171577

CHANGE HISTORY

Version	Date
1.0	15.04.2019
1.1	26.04.2019

Table Of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose of the System

Purpose of Smart METU Campus is to make users (which in our case are students, instructors and workers) daily life in METU Campus easier, such as reserving spots in library, reporting problems, seeing available books online.

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:

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

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 Stakeholders and their concerns

Students: Students are current METU Students. Their main concerns are to able to reserve a spot on time and to report problems regarding computers directly. For this to work smoothly, they should have a clean interface and 24/7 working servers.

Instructors: Instructors are current METU instructors. Their primary concern is being able to report problems easily. This requires a clean interface so that they can give details of the problems with no ambiguity. Also viewing existing problems is an important concern so that they won't create an existing problem.

Workers: Workers are people who are hired from METU administration to solve problems such as Wi-Fi and computer problems. Their primary concern is viewing reported problems and being able to understand them easily. This requires a clean interface so that there is no ambiguity. Also they need to easily mark the reported problems as solved.

System Admins: System admins are the people who is in charge of maintaining the system also responsible of blocking users who doesn't act accordingly. Their main concerns are to provide working servers 24/7 with exception of 5 minutes per year and solving reported account problems.

Software Developers: Software developers are people who are hired by METU administration to develop Smart METU Campus system. Their main concern is providing METU administration and system users needs with given resources. They may need to convince METU administration if the requested task is not viable with given resources.

METU Administration: METU Administration is the people who wants to have this system and gives ,3resources for it. Their main concern is to have their requests met in a best way possible with the resources they have.

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 Glossary

Term	**Definition**
Student	Student at METU who has an account on the system.
Instructor	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.
Admins	Expert who manages the entire system.
SATA	Serial Advanced Technology Attachment. An interface for transferring data between a computer and a storage device.
SAS	Serial Attached Small Computer System Interface. An interface for transferring data between a computer and a storage device.
HTTP Protocol	HTTP is a protocol which allows the fetching of resources, such as HTML documents.
MongoDB	MongoDB is an object-oriented, NoSQL database and it usesJSON to store data.
JSON	JSON is a syntax for storing and exchanging data. and it is written with JavaScript object notation.

Table 1: Glossary

4 Architectural Views

4.1 Context View

In this viewpoint, context of the system with all actors are defined in general and detailed viewpoints. In the context diagram, actors and their interaction with the Smart METU Campus System will be explained in general terms. Use case diagrams and the detailed explanations of every possible use case of the system will be specified below the use case diagram.

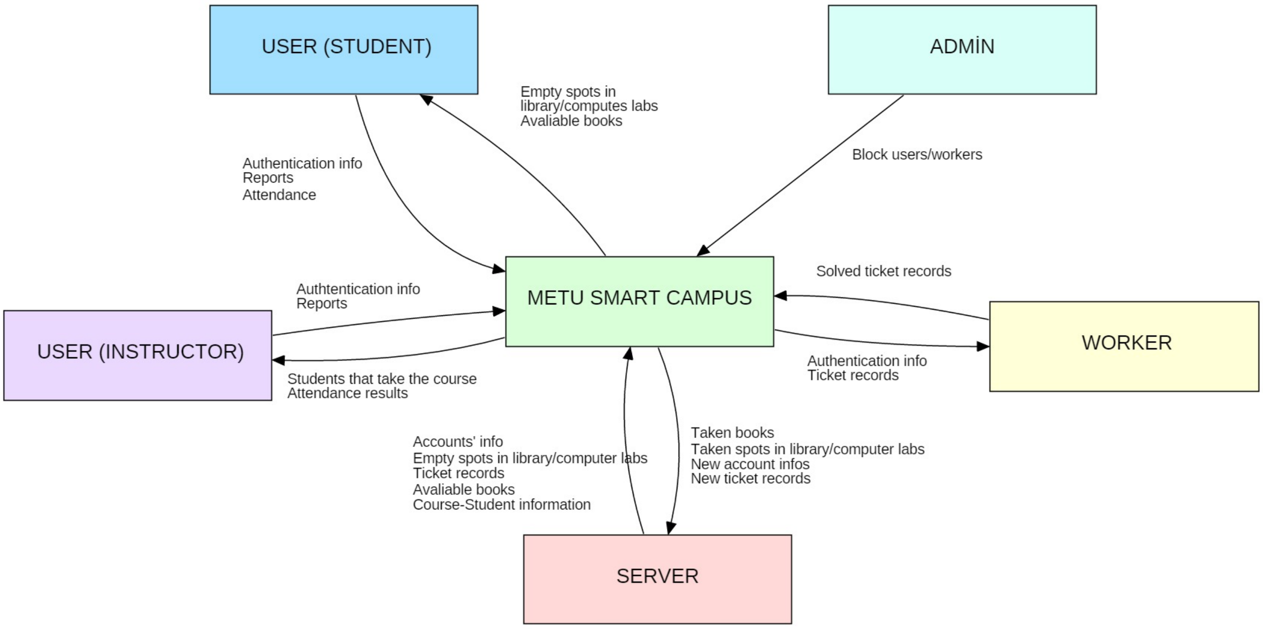

Figure 1: Context Diagram

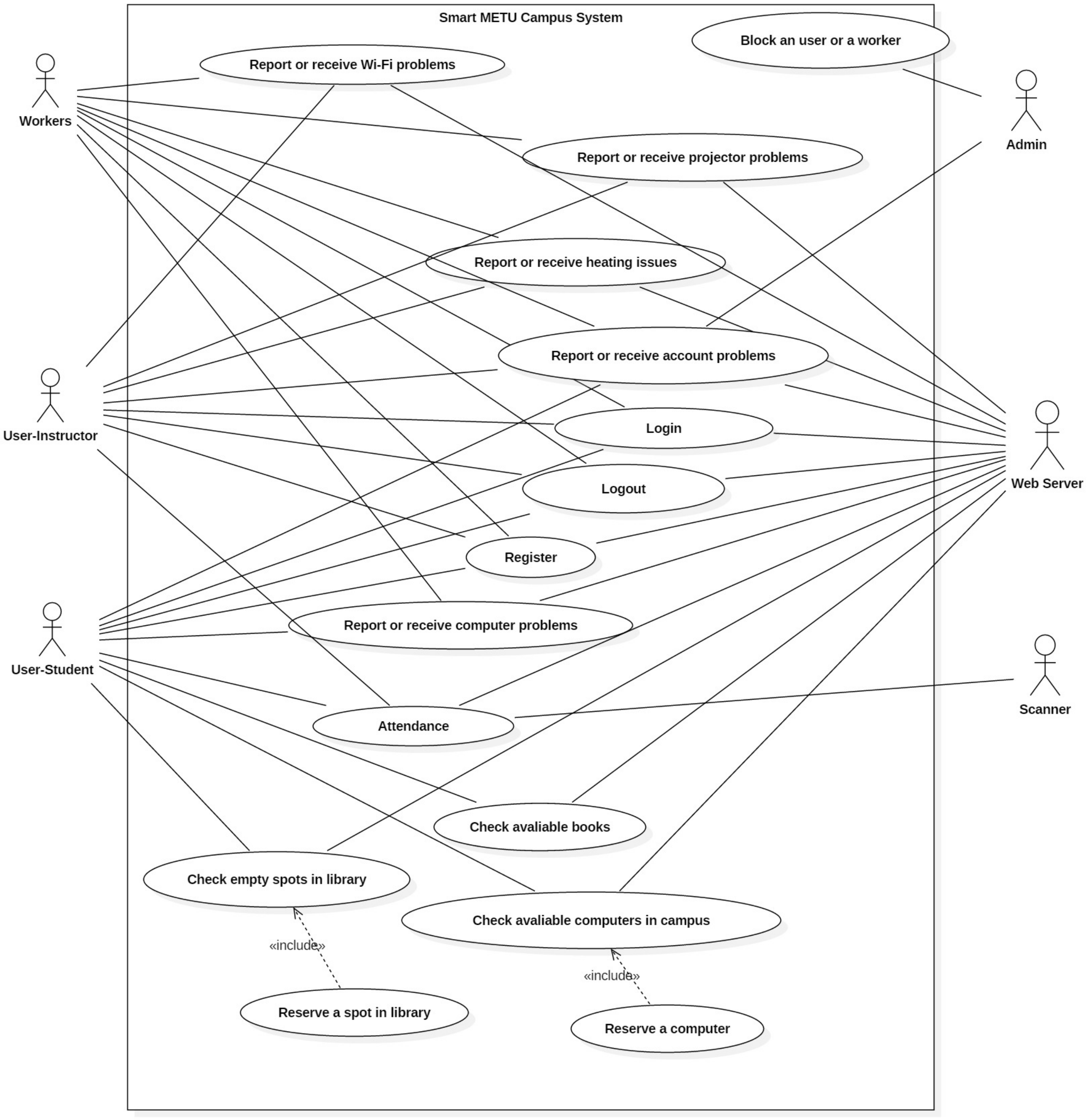
Smart METU Campus System
Block an user or a worker
Report or receive Wi-Fi problems
Workers
Admin
Report or receive projector problems
Report or receive heating issues
Report or receive account problems
User-Instructor
Login
Web Server
Logout
Register
Report or receive computer problems
User-Student
Scanner
Attendance
Check avaliable books
Check empty spots in library
Check avaliable computers in campus
«include»
«include»
Reserve a spot in library
Reserve a computer

Figure 2: Use Case Diagram

Use Case ID	**1**
Use Case Name	Report heating problems
Actors	Users(Instructors)
Description	In case of a heating problem in a classroom instructors can report it to workers.
Data	reportID, reportDescription, userID
Pre-conditions	1. Instructor should log in to the system. 2. Instructor should fill the details about the problem.
Stimulus	Instructor presses ”Create a Report” button.
Normal Flow	1. Instructor logs in to the system. 2. Instructor presses ”Create a Report” button. 3. Intsructor fills problems details and presses ”Submit” button.
Alternative Flow	3. Problem is already created by another instructor.
Exceptions	Instructor cannot log in to the system due to connection lost.
Post-conditions	Ticket is created and is shown in the workers’ interface.

Table 2: Use case description for reporting heating problems

Use Case ID	**2**
Use Case Name	Receive heating problems
Actors	Workers
Description	In case of a heating problem in a classroom is reported from intructors, workers can see it in app’s interface.
Data	reportID, reportDescription, userID
Pre-conditions	1. Workers should log in to the system. 2. At least a problem should be reported.
Stimulus	Worker selects ”Show reports” from the dropdown menu.
Normal Flow	1. Worker logs in to the system. 2. Worker selects ”Show Reports” from the menu. 3. Worker views the reports.
Alternative Flow	3. There is no heating problem reported.
Exceptions	Worker cannot log in to the system due to connection lost.
Post-conditions	Worker is informed about the heating problems.

Table 3: Use case description for receiving heating problems

Use Case ID	3
Use Case Name	Report Computer Problems
Actors	Students, Workers
Description	In case of a computer problem in a computer lab students can report it to workers.
Data	StudentID, problem description
Preconditions	1. Student should log into the system. 2. Student should navigate to report problems section.
Stimulus	Student chooses the computer to be reported and clicks report button
Basic Flow	1. Student logs in the system. 2. Student navigates to report problems section. 3. Student chooses computer problem from the dropdown menu. 4. Student chooses the computer to be reported. 5. Student clicks the report button.
Alternative Flow	-
Exception Flow	If the connection is lost, the chosen computer should be stored in local memory.
Postconditions	Workers is informed about computer problem.

Table 4: Use case description for reporting computer problems

Use Case ID	4
Use Case Name	Reciving Computer Problems
Actors	Workers
Description	In case of a computer problem in a computer lab is reported from students, workers can see it in app's interface.
Data	WorkerID, problem description
Preconditions	1. Worker should log into the system. 2. Worker should navigate to report problems section.
Stimulus	Worker selects ”Show reports” from the dropdown menu.
Basic Flow	1. Worker logs in the system. 2. Worker selects ”Show Reports” from the menu. 3. Worker views the reports.
Alternative Flow	3. There is no computer problem reported.
Exception Flow	Worker cannot log in to the system due to connection lost.
Postconditions	Worker is informed about the heating problems.

Table 5: Use case description for receiving computer problems

Use Case ID	**5**
Use Case Name	Report Wi-Fi problems
Actors	Users(Instructors)
Description	In case of a Wi-Fi connection problem instructors can report it to workers.
Data	reportID, reportDescription, userID
Pre-conditions	1. Instructor should log in to the system. 2. Instructor should fill the details about the problem.
Stimulus	Instructor presses "Create a Report" button.
Normal Flow	1. Instructor logs in to the system. 2. Instructor presses "Create a Report" button. 3. Intsructor fills problems details and presses "Submit" button.
Alternative Flow	3. Problem is already created by another instructor.
Exceptions	Instructor cannot log in to the system due to connection lost.
Post-conditions	Ticket is created and is shown in the workers' interface.

Table 6: Use case description for reporting Wi-Fi problems

Use Case ID	**6**
Use Case Name	Receive Wi-Fi problems
Actors	Workers
Description	In case of a Wi-Fi connection problem is reported from intructors, workers can see it in app's interface.
Data	reportID, reportDescription, userID
Pre-conditions	1. Worker should log in to the system. 2. At least a problem should be reported.
Stimulus	Worker selects "Show reports" from the dropdown menu.
Normal Flow	1. Worker logs in to the system. 2. Worker selects "Show Reports" from the menu. 3. Worker views the reports.
Alternative Flow	3. There is no Wi-Fi connection problem reported.
Exceptions	Worker cannot log in to the system due to connection lost.
Post-conditions	Worker is informed about the Wi-Fi problems.

Table 7: Use case description for receiving Wi-Fi problems

Use Case ID	**7**
Use Case Name	Report projector problems
Actors	Users(Instructors)
Description	In case of a projecor problem in a classroom instructors can report it to workers.
Data	reportID, reportDescription, userID
Pre-conditions	1. Instructor should log in to the system. 2. Instructor should fill the details about the problem.
Stimulus	Instructor presses ”Create a Report” button.
Normal Flow	1. Instructor logs in to the system. 2. Instructor presses ”Create a Report” button. 3. Intsructor fills problems details and presses ”Submit” button.
Alternative Flow	3. Problem is already created by another instructor.
Exceptions	Instructor cannot log in to the system due to connection lost.
Post-conditions	Ticket is created and is shown in the workers’ interface.

Table 8: Use case description for reporting projector problems

Use Case ID	**8**
Use Case Name	Receive projector problems
Actors	Workers
Description	In case of a projector problem in a classroom is reported from instructors, workers can see it in app’s interface.
Data	reportID, reportDescription, userID
Pre-conditions	1. Worker should log in to the system. 2. At least a problem should be reported.
Stimulus	Worker selects ”Show reports” from the dropdown menu.
Normal Flow	1. Worker logs in to the system. 2. Worker selects ”Show Reports” from the menu. 3. Worker views the reports.
Alternative Flow	3. There is no projector problem reported.
Exceptions	Worker cannot log in to the system due to connection lost.
Post-conditions	Worker is informed about the projector problems.

Table 9: Use case description for receiving projector problems

Use Case ID	9
Use Case Name	Reserve a spot in library
Actors	Students
Description	If there is an empty spot in library, student can reserve this spot for 5 minutes until he/she gets there.
Data	StudentID, empty_spots, reserved_spots
Preconditions	1. Students should log into the system. 2. There has to be an empty spot in library.
Stimulus	Student logins into the system, sees empty spots, reserves it for 5 minutes
Basic 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.
Exception Flow	If there is no empty spot so student cannot reserve a spot.
Postconditions	1. Spot is reserved for 5 minutes. 2. If the student doesn't get there on time spot will be seen as empty again.

Table 10: Use case description for reserving a spot in library

Use Case ID	10
Use Case Name	Reserve a spot in computer lab
Actors	Students
Description	If there is an empty computer in computer lab, student can reserve this computer for 5 minutes until he/she gets there.
Data	StudentID, empty_computers, reserved_computer
Preconditions	1. Students should log into the system. 2. There has to be an empty spot in computer library.
Stimulus	Student logins into the system, sees empty computers, reserves it for 5 minutes
Basic Flow	1. Student logs in the system. 2. There is an empty computer that can be reserved. 3. Student reserves the computer and the spot is computer as busy for 5 minutes. 4. Student gets there and takes the spot.
Alternative Flow	4. Student doesn't take the computer on time, computer is displayed as empty again and the student has to wait for an hour to reserve a spot.
Exception Flow	If there is no empty computer, student cannot reserve a computer.
Postconditions	1. Computer is reserved for 5 minutes. 2. If the student doesn't get there on time copmuter will be seen as empty again.

Table 11: Use case description for reserving a spot in computer lab

Use Case ID	**11**
Use Case Name	Report account problems
Actors	Users(Students, Workers, Instructors)
Description	In case of a projector problem in a classroom is reported from instructors, workers can see it in app's interface.
Data	reportID, reportDescription, userID
Pre-conditions	1. User should log in to the system. 2. User should fill the details about the problem.
Stimulus	User presses "Create a Report" button.
Normal Flow	1. User logs in to the system. 2. User presses "Create a Report" button. 3. Intsructor fills problems details and presses "Submit" button.
Alternative Flow	-
Exceptions	User cannot log in to the system due to connection lost.
Post-conditions	Ticket is created and is shown in the admin's interface.

Table 12: Use case description for reporting account problems

Use Case ID	**12**
Use Case Name	Receive account problems
Actors	Admins
Description	In case of a account problem is reported from users, admins can see it in app's interface.
Data	reportID, reportDescription, userID
Pre-conditions	1. Admin should log in to the system. 2. At least a problem should be reported.
Stimulus	Admin selects "Show reports" from the dropdown menu.
Normal Flow	1. Admin logs in to the system. 2. Admin selects "Show Reports" from the menu. 3. Admin views the reports.
Alternative Flow	3. There is no account problem reported.
Exceptions	Admin cannot log in to the system due to connection lost.
Post-conditions	Admin is informed about the account problems.

Table 13: Use case description for receiving account problems

Use Case ID	**13**
Use Case Name	Block an user
Actors	Admin
Description	In case of an user violates rules and regulations admin can block the user activities.
Data	userID
Pre-conditions	1. Admin should log in to the system.
Stimulus	Admin selects ”Block the user” option.
Normal Flow	1. Admin logs in to the system. 2. Admin selects ”Block the user” option.
Alternative Flow	-
Exceptions	Admin cannot log in to the system due to connection lost.
Post-conditions	Admin blocks the user’s activities.

Table 14: Use case description for blocking an user

Use Case ID	**14**
Use Case Name	Check available books
Actors	User(Student)
Description	Students can check available books in library.
Data	userID, bookID
Pre-conditions	1. Student should log in to the system.
Stimulus	Student writes the name of the book in the search bar.
Normal Flow	1. Student logs in to the system. 2. Student selects ”Search in Library” option. 3. Student writes the name of the book in the search bar.
Alternative Flow	-
Exceptions	Student cannot log in to the system due to connection lost.
Post-conditions	Student is informed about the availability of the book.

Table 15: Use case description for checking available books

Use Case ID	15
Use Case Name	Attendance
Actors	Students, Instructor, Scanner
Description	Students scans their METU student ID card in the classroom, attendance data displays on the instructor's computer and instructor confirms it.
Data	StudentID, InstructorID, attendance_data
Preconditions	1. Students should be in classroom on time to be able to scan their IDs. 2. Instructor should login into the system. 3. Instructor should navigate to where the attendance information is
Stimulus	Students scan their IDs, Instructors logins to the system and confirms the attendance.
Basic Flow	1. Students scan their IDs to the scanner which is located in every desk in front of the seats. 2. Instructor logins to the system. 3. Instructor navigates to where the attendance information is 4. Instructor confirms the attendance if It is true.
Alternative Flow	4. If a seat is empty which is seen to be full by a certain student, Instructor, specifies that student is not here. 4. If a seat is seen to be full by an another student, Instructor specifies who is in the class and who is not.
Exception Flow	Scanner does not work.
Postconditions	Instructor is informed about the attendance.

Table 16: Use case description for Attendance

4.2 Composition View

In this viewpoint, top-level decomposition of project Smart METU Campus with its components and functions,interfaces of each one of those parts are explained in detail with diagrams.

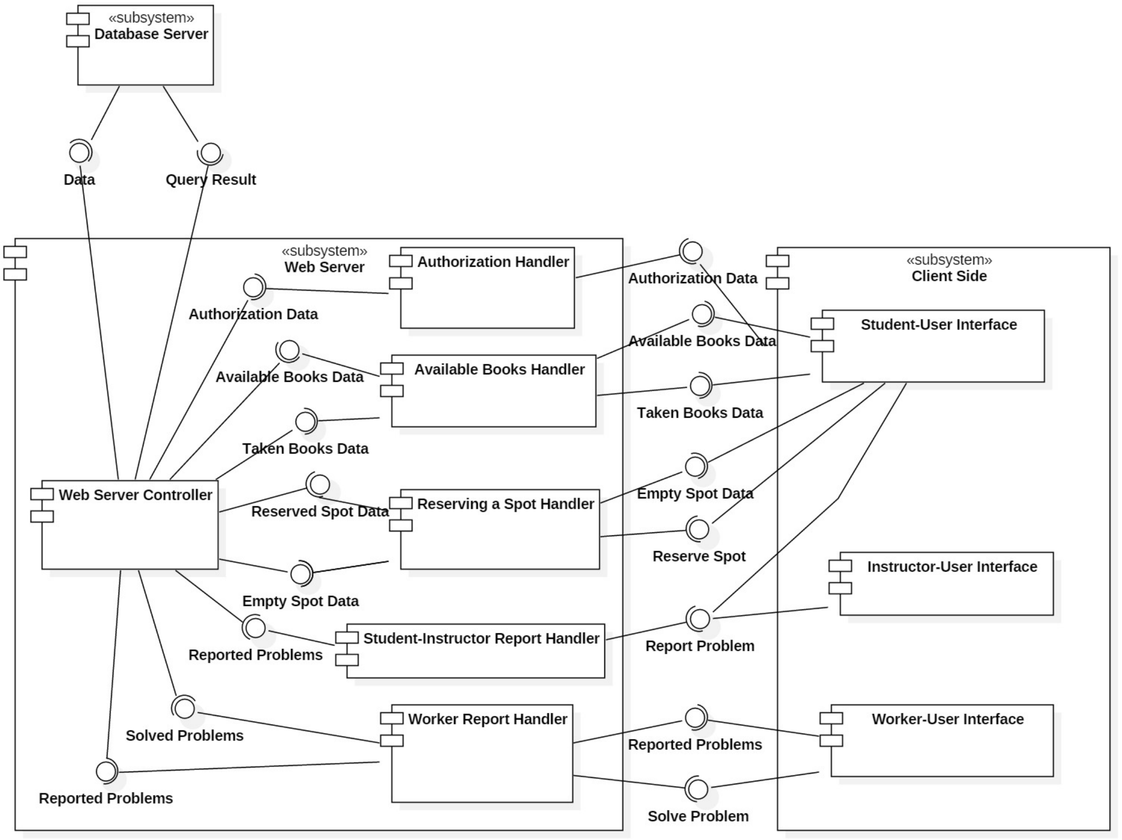

Figure 3: Component Diagram

Design Rationale:

- Web Server component consists of different components that handle user-system interaction.
- Database Server component is responsible for storing data(empty/reserved spots, available books, users' information etc.) and handing requested data to the Web Server Controller component.
- Web Server Controller component gives/gets the data from database and serves/gets

the necessary data to the related components. For example it takes the empty spots data from database and gives it to the Reserving a Spot Handler component. Then it takes the reserved spots data from again Reserving a Spot Handler component and gives it back to the database.

- Authorization Handler is responsible for authorizing users(students,instructors, workers) for Smart METU Campus system.
- Worker Report Handler component gets reported problems from database through Web Server Controller component, serves them to workers. Then gets the solved problems from workers and notifies the system.
- Student-Instructor Report Handler transfers reported problems to the database through Web Server Component.
- Available Books Handler component transfer the data about books availability.
- Reserve a Spot Handler component sends reserved spots data and receives empty spots data.
- Client Side component consists 3 different user-interface components.
- In the Client Side all the users have the same sign up and login page where users type their credentials.
- Student-User Interface is where students see available books, empty spots and reserve a spot, free a spot and also report a problem.
- Worker-User Interface is where workers see reported problems and check them as solved.
- Instructor-User Interface is where instructors report problems.

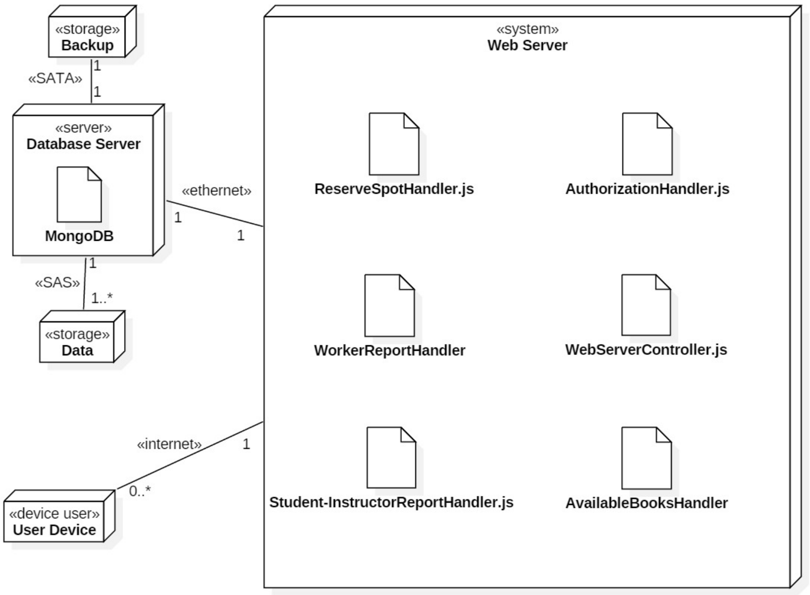

Figure 4: Deployment Diagram

Design Rationale:

- Web server is developed and running with JavaScript. Thus we use MongoDB in database management. MongoDB is an object-oriented, NoSQL database and it uses JSON to store data. Also MongoDB provides compatibility between front-end and back-end since we use Node.js (which allows us to run JavaScript on the back-end) to connect database and interface.
- Users will communicate with web server with internet connection.
- We use a local database which is connected to the web server with ethernet.
- We use SATA for backup since it is more suitable for backup storing.
- We use SAS for storing data since it is faster and more reliable than SATA.
- All internet based connections will use encrypted communication protocols for increased security (SSL,HTTPS).

4.3 Information View

In this view, the organization and the relations of the data that will be stored with the operations of the system that create, use, modify and delete the data will be specified. Also, the effects of the system operations on the data will be examined in terms of their effect type; create, read, update and delete.

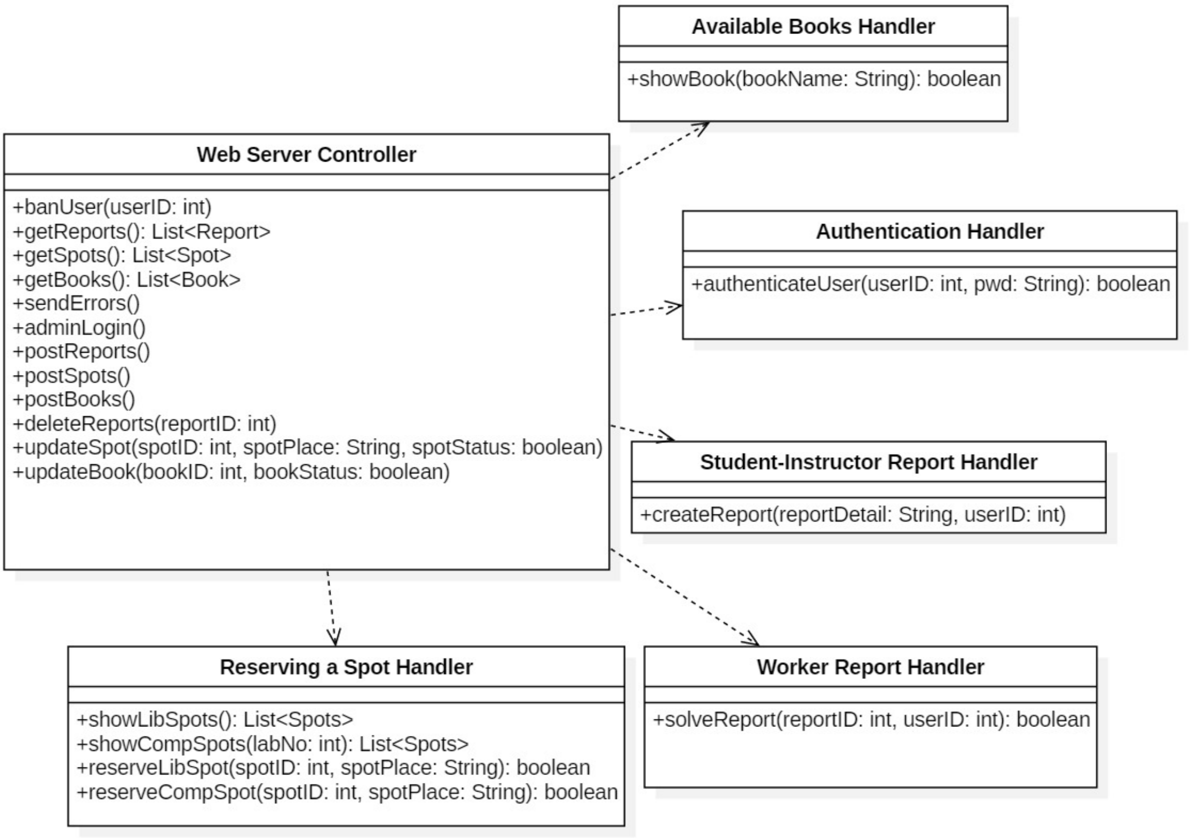
Available Books Handler
+showBook(bookName: String): boolean
Web Server Controller
+banUser(userID: int)
+getReports(): List<Report>
+getSpots(): List<Spot>
+getBooks(): List<Book>
+sendErrors()
+adminLogin()
+postReports()
+postSpots()
+postBooks()
+deleteReports(reportID: int)
+updateSpot(spotID: int, spotPlace: String, spotStatus: boolean)
+updateBook(bookID: int, bookStatus: boolean)
Authentication Handler
+authenticateUser(userID: int, pwd: String): boolean
Student-Instructor Report Handler
+createReport(reportDetail: String, userID: int)
Reserving a Spot Handler
+showLibSpots(): List<Spots>
+showCompSpots(labNo: int): List<Spots>
+reserveLibSpot(spotID: int, spotPlace: String): boolean
+reserveCompSpot(spotID: int, spotPlace: String): boolean
Worker Report Handler
+solveReport(reportID: int, userID: int): boolean

Figure 5: Service Interfaces Class Diagram

4.3.1 CRUD Operations

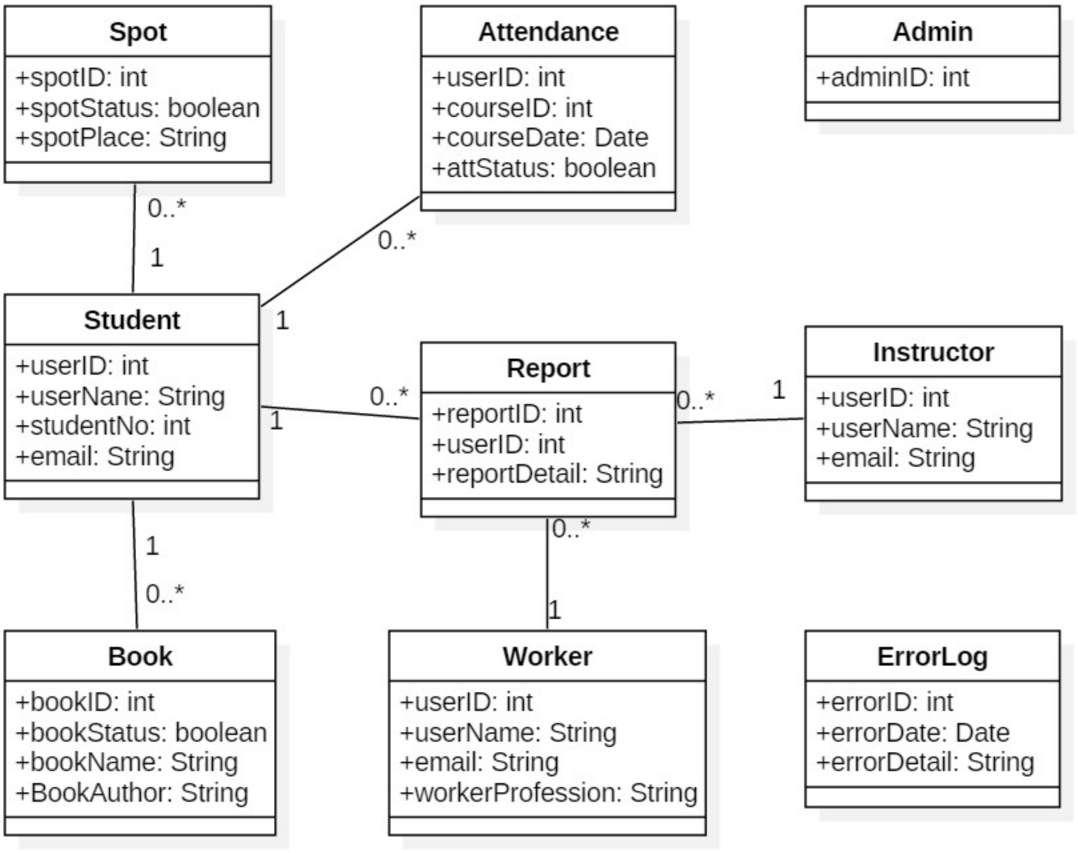

Figure 6: Database Class Diagram

Operation	CRUD Operations

getReports	CREATE -
	READ - Report
	UPDATE -
	DELETE -

getSpots	CREATE -
	READ - Spot
	UPDATE -
	DELETE -

getBooks	CREATE -
	READ - Book
	UPDATE -
	DELETE -

sendErrors	CREATE -
	READ - ErrorLog
	UPDATE -
	DELETE -

Operation	CRUD Operations

deleteReports	CREATE -
	READ -
	UPDATE -
	DELETE - Report

updateSpots	CREATE -
	READ -
	UPDATE - Spot
	DELETE -

updateBook	CREATE -
	READ -
	UPDATE - Book
	DELETE -

Design Rationale:

- All the information about users, reports, books and spots are hold in the database.
- Attendance is a weak entity since It cannot exists without student.
- Report is a weak entity since It cannot exists without, students and instructors.
- Students can update Spot and Book but they cannot add any book or spot.

4.4 Interface View

In this view, the internal interfaces between the components of the system and the external interfaces of Smart METU Campus System will be specified in detail.

4.4.1 Internal Interfaces

The Interface Between the Database Server and the Web Server Controller

The Web Server Controller queries the database when an operation in the system need a certain data. The query is in object from of JavaScript. The database run query using MongoDB. If query is failed the error message would be sent back to the Web Server Controller. If query is successful, the result would be sent to the Web Server Controller.

Design Rationale:

- The system is all based on data of users, reports, spots etc. The interface that communicates with database and distributes the data to the other interfaces is needed.

The Interface Between the Authentication Handler and Web Server Controller

In order to let user log in to the system and validate the user's login information there is a need for an interface which sends the log in info to the database and validates the result.

Design Rationale:

- Authentication Handler cannot communicate with database directly. It has to send the log in data to the database through Web Server Controller. Thus an interface between these components is needed.

The Interface Between the Available Books Handler and Web Server Controller

When a student wants to reach books' information the requested book information is sent to the Web Server Controller.

Design Rationale:

- Available Books Handler cannot communicate with the database directly. It has to send the books' information to the database through Web Server Controller. Thus an interface between these components is needed.

The Interface Between the Reserving a Spot Handler and Web Server Controller

When a student wants to check status of spots in library/computer labs the request is sent to the Web Server Controller.

Design Rationale:

- Reserving a Spot Handler cannot communicate with the database directly. It requests the data of spots from database through Web Server Controller. Thus an interface between these components is needed.

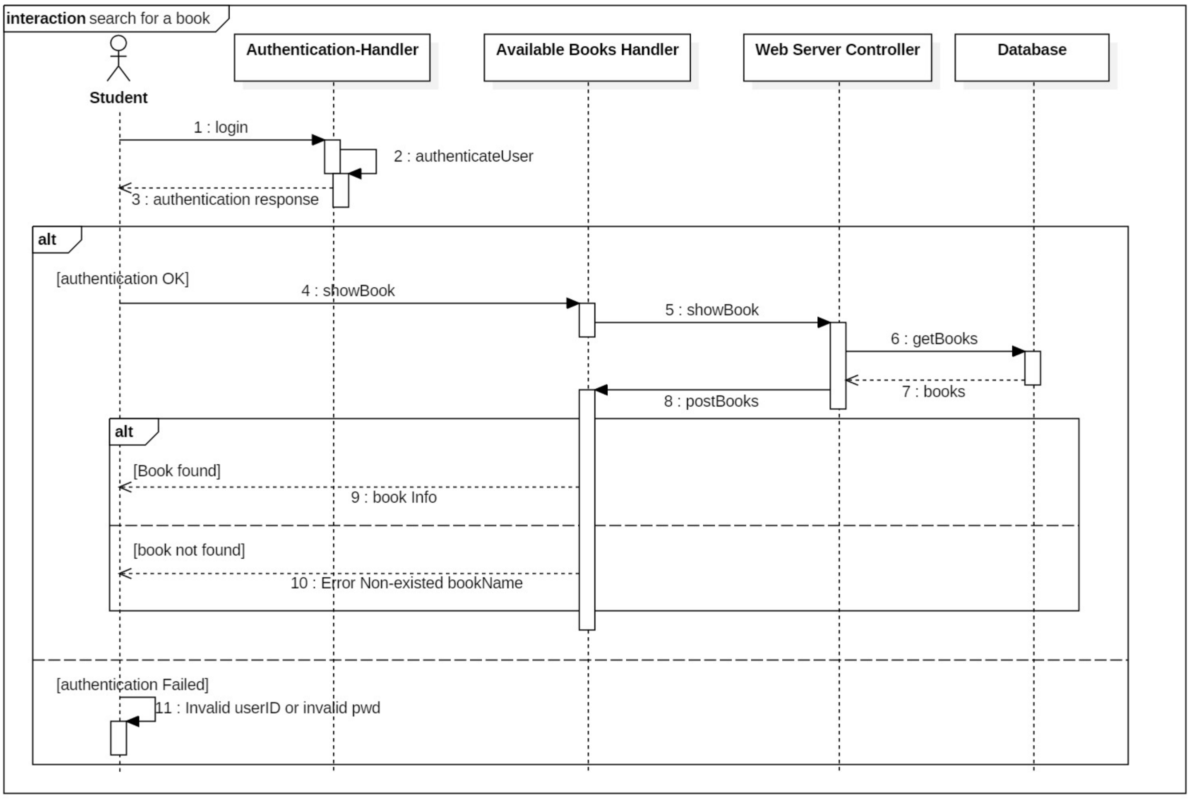

Figure 7: Search for a Book sequence diagram showing the interfaces between internal interfaces Authentication Handler, Available Books Handler and Web Server Controller

The Interface Between the Student-Instructor Report Handler and Web Server Controller

When a student or an instructor wants to report a problem, the details of the problem is sent to the Web Server Controller.

Design Rationale:

- Student-Instructor Report Handler cannot communicate with the database directly. It has to send the data that is going to be written to the database to the Web Server Controller. Then Web Server Controller sends the report to the database. Thus an interface between these components is needed.

The Interface Between the Worker Report Handler and Web Server Controller

When a worker wants to check the problems that are reported the handler sends the request to the Web Server Controller. Or when a worker wants to mark a problem as solved so delete from database, the handler send the report data and deletion request to the Web Server Controller.

Design Rationale:

- Worker Report Handler cannot communicate with the database directly. It sends the request of viewing reports or the report's info that needs to be deleted from the database to the Web Server Controller. Then the Web Server Controller sends these requests to the database. Thus an interface between these components is needed.

4.4.2 Service Interfaces

Operation	Description
banUser	Admin can suspend a user with the given user's ID from METU smart campus system
getReports	Gets the reports from the database.
postReports	Post reports to the worker report handler.
getSpots	Gets the spots' status and information from the database.
postSpots	Post the spots' status and information to the reserving a spot handler.
getBooks	Gets the books' status and information from the database.
sendErrors	Send the system errors to the admin.
adminLogin	Let's the admin login to the system.
deleteReports	Delete reports from the database.
updateSpots	Update the spots' status in the database.
updateBook	Update the books' status in the database.
solveReports	From worker report handler to web server controller, inform that the report is going to be deleted from database.
showLibSpots	Send library spots' status amd information to the student's interface.
showCompSpots	Send computer lab spots' status amd information to the student's interface.
reserveLibSpot	Take reserved library spot's information and send it to the web server controller.
reserveCompSpot	Take reserved computer lab spot's information and send it to the web server controller.
authenticateUser	Send user's login info and validate it.
createReport	Create a report from given information in the user interface.
showBook	Show book's status and information to the user interface.

Table 17: Service Interfaces Operation Descriptions

Operation	**Inputs**	**Outputs**	**Exceptions**
banUser	-userID	True if operation was successful, otherwise false	- Database connection error occurs. - Invalid userID.
getReports	-	List of reports in the database.	- Database connection error occurs.
postReports	-	True if operation was successful, otherwise false	- Database connection error occurs
getSpots	-	List of spots.	- Database connection error occurs
postSpots	-	True if operation was successful, otherwise false	- Database connection error occurs
getBooks	-	List of books in the database.	- Database connection error occurs
sendErrors	-	True if operation was successful, otherwise false	- Connection error with network occurs
adminLogin	-	True if operation was successful, otherwise false	- Connection error with network occurs
deleteReports	-reportID	True if operation was successful, otherwise false	- Database connection error occurs - Invalid reportID
updateSpots	-spotID -spotStatus -spotPlace	True if operation was successful, otherwise false	- Database connection error occurs - Invalid spotID.
updateBook	-bookID -bookStatus	True if operation was successful, otherwise false	- Database connection error occurs. - Invalid bookID
solveReports	-reportID -userID	True if operation was successful, otherwise false	- Database connection error occurs. - Invalid reportID or invalid userID.
showLibSpots	-	List of library spots in the database.	- Database connection error occurs
showCompSpots	-	List of computer lab spots in the database.	- Database connection error occurs
reserveLibSpot	-spotID -spotPlace	True if operation was successful, otherwise false	- Database connection error occurs. - Invalid spotID.
reserveCompSpot	-spotID -spotPlace	True if operation was successful, otherwise false	- Database connection error occurs. - Invalid spotID.
authenticateUser	-userID -pwd	True if operation was successful, otherwise false	- Connection error with network occurs. - Invalid userID or invalid pwd
createReport	-reportDetail -userID	True if operation was successful, otherwise false	- Database connection error occurs. - Invalid userID or empty reportDetail
showBook	-bookName	The requested book's availability in boolean.	- Database connection error occurs. - Non-existed bookName.

Table 18: Operation Design

Design Rationale:

- There are various handler classes to communicate web server controller. with the client side interfaces. This is because firstly, there are many client side interfaces it is easier to handle with smaller modules. Secondly, there are already many operations that web server controller handles.
- Only web server controller can access to database this way it is easier to manage and it is more secure.

4.4.3 External Interfaces

4.4.3.1 User Interfaces

Login page for all the user groups which are students, instructors and workers are the same but once users login, a different page comes up depending on their user group and their personal information.

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. The last one is where they can see their attendance information.

Design Rationale:

- On the right side of the page, students name, email and student number is displayed.
- Spots are shown as dropdown menu.
- Next to each spot there is green or red box, green box indicating the spot is empty, red box indicating the spot is full.
- If the box next to spot is green there is also reserve button next to it.
- If student doesn't get to spot in 5 minutes, reserve button for all the seat is unabled for an hour.

- There is a empty box next to search button for books. After typing a book and click search button, new page appears, displaying if the book is in the library or not.
- Attendance information is shown in a new page after clicking attendance information button. In the new page there is list of every courses student is taking and information about how many days they were absent.
- Problems are shown as dropdown menu.
- After selecting one problem to report, report button appears next to it.
- After reporting a problem a popup screen appears declaring that they successfully reported the problem.

Worker-User Interface

The user-interface for the workers has one main component where they can see the problems.

Design Rationale:

- On the right side of the page, workers name, email and profession is displayed.
- Problems are shown as a list.
- If a reported problem is selected to be check as solved, a solved as checked button appears next to it, if worker clicks that button, problem is deleted from the list.

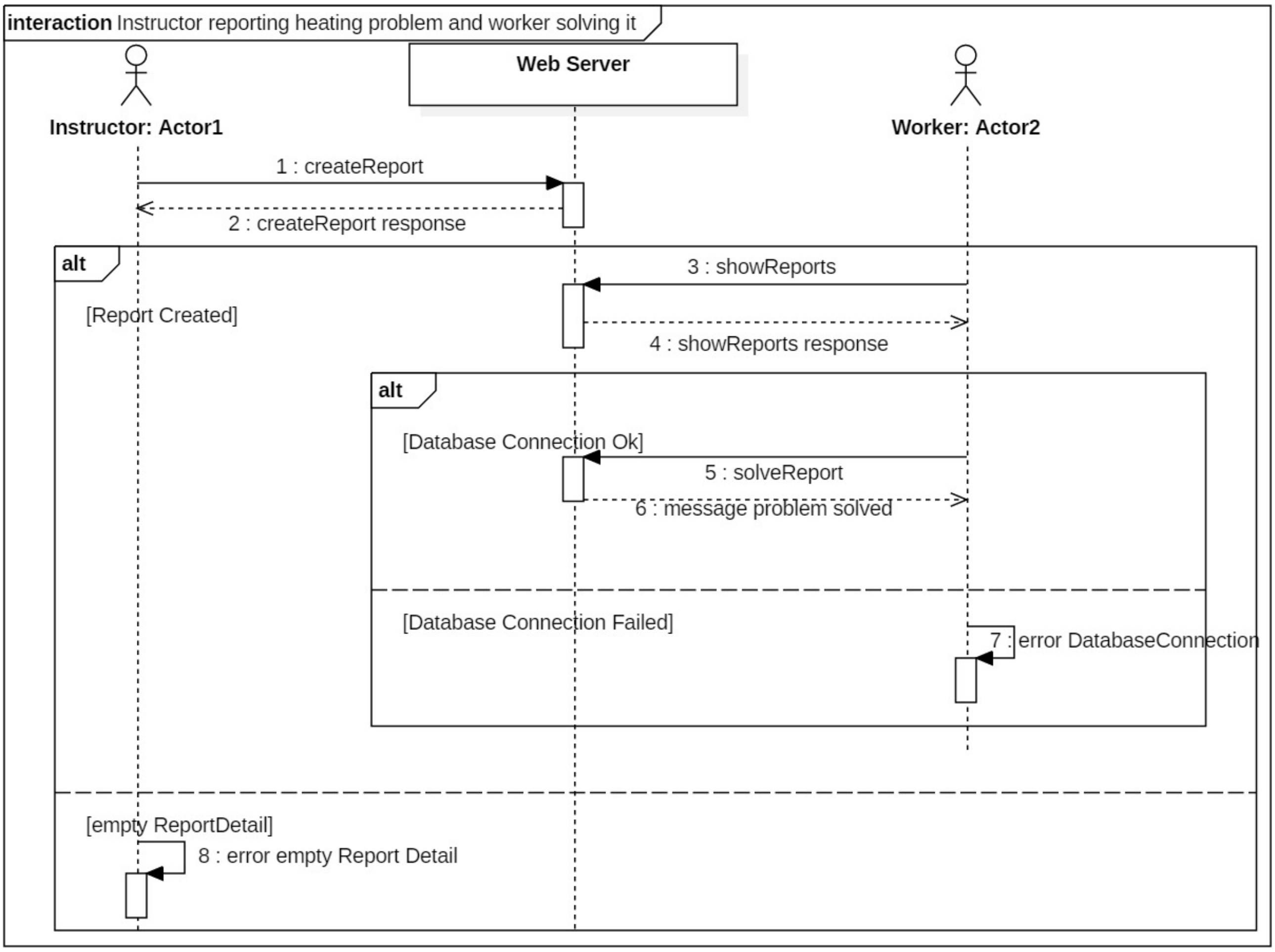
interaction Instructor reporting heating problem and worker solving it
Web Server
Instructor: Actor1
Worker: Actor2
1 : createReport
2 : createReport response
alt
3 : showReports
[Report Created]
4 : showReports response
alt
[Database Connection Ok]
5 : solveReport
6 : message problem solved
[Database Connection Failed]
7 : error DatabaseConnection
[empty ReportDetail]
8 : error empty Report Detail

Figure 8: Sequence diagram of Instructor reporting a heating problem and worker solving it

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.

Design Rationale:

- On the right side of the page, instructors name, email is displayed.
- Problems are shown as dropdown menu.
- After selecting one problem to report, report button appears next to it.
- After reporting a problem a popup screen appears declaring that they successfully

reported the problem.

- To be able to see the attendance information, instructor should click the attendance button.
- After clicking the attendance information, current classes' bird's eye view appears in the screen with the students (who have scanned their card to scanner on their seat) name, and student no in each seat.
- If the attendance is not correct, teacher can change it clicking the seats.
- After confirming attendance pop up screen appears to declare that they successfully confirmed the attendance.

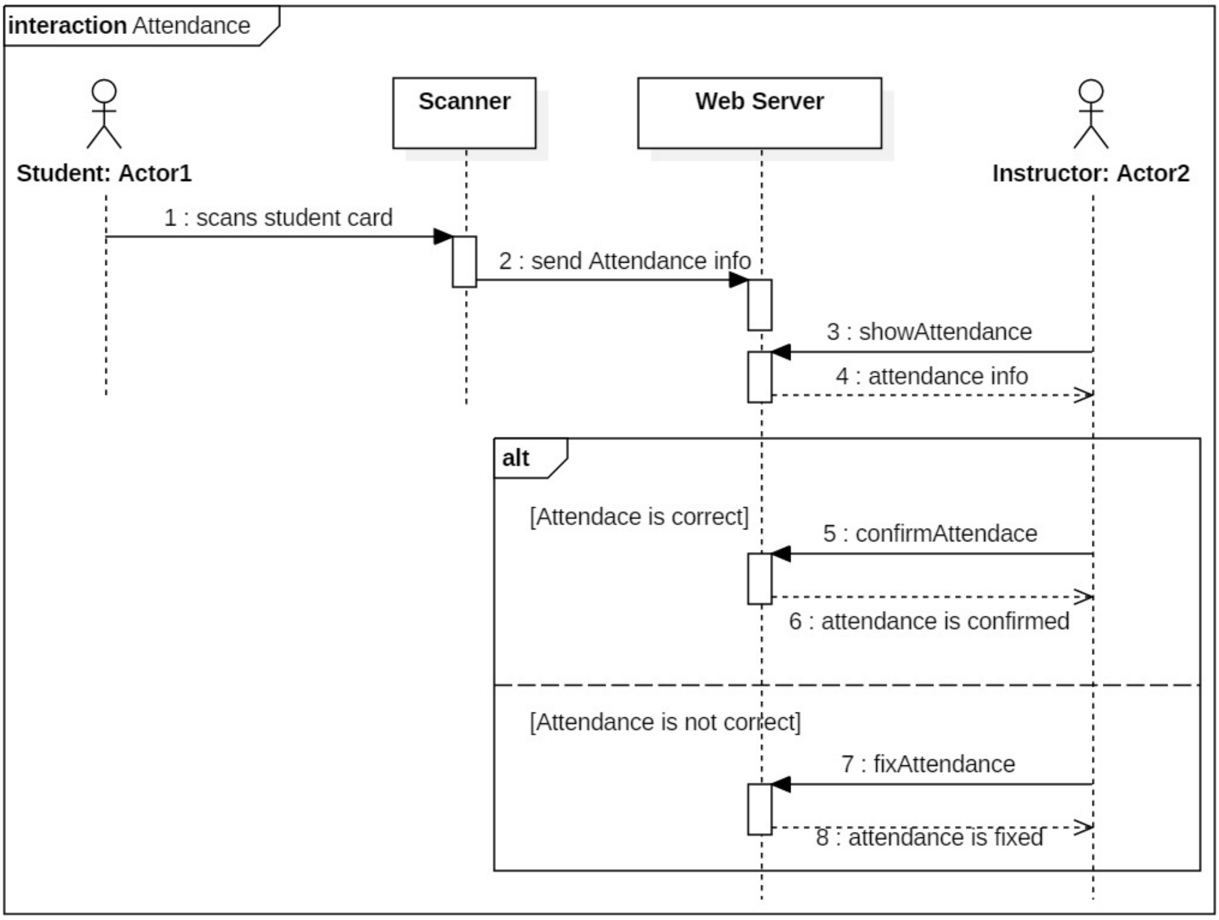

Figure 9: Sequence diagram of student participating in attendance and instructor confirming it