Smart METU Campus

Software Design Descriptions for the Smart METU Campus Project

PROJECT EXECUTING TEAM

Project Member	METU Number	Signature
Kürşat Öztürk	e2171874	
Beste Burhan	e2171395	

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Purpose of the System

This project aims to handle daily problems easily that students and instructors face in daily life of university. Such problems are knowing ring location information and time it arrives to the stations, seeing and reserving available seats & books in the library, and attendance to the classes.

1.2 Scope

- Project will include an interface that provides students, instructors and staff to see ring information. Not only exact location, but also the approximate time information that ring will arrive to the stations via web application.
- Project will include an interface that provides students and instructors to see and reserve available books and seats in library via an IoT located in the entrance of library or web application.
- Project will include an interface that provides to students that they can give attendance to the classes they attend.
- Project will provide instructors to take attendance in an easy and without any cheating, that often occurs in schools.
- System will hold attendance information in database so that if course syllabus requires it, instructor giving the course can check students' attendance information at the end of semesters.
- System will hold a login interface for users. Users can login either via reading ID Card to certain IoT devices or entering the username & password of ODTU system created when they registered to the school.

1.3 Stakeholders and their Concerns

Users: There will be three type of users in this system. Students, Instructors and Staffs. Students and instructors are concerned about library utilities usage order. In this system, they will be able to reserve either seat or books for themselves in order without any conflict. Also, they will be able to track their attendance information with ease. Instructors are mainly concerned about attendance of students to the classes. With this system, attendance information will be gathered and easily tracked by instructor. All users are interested in knowing ring locations, and this will be available in the system. Although, there will be no critical user information stored or used other than user credential and METU Card information in the system, privacy is main concern for users. Staff has no other concerns.

IT Staff: IT Staff are the people who is responsible for maintaining the system's health. For any kind of failure in system must reach them in well-defined tickets. And they are concerned with the interface provided them to interact with system is easy-to-use and has all capabilities necessary and permissions as admin.

System Developers: They are the group who are developing the system. Their concern is well-defined requirements and limited constraints. They wish personals who has no idea about constructing such a system will not interfere the development process.

2. References

This document is written with respect to IEEE 1016-2009 standard:

IEEE. (2009, July 20). 1016-2009 Standard for Information Technology—Systems Design—Software Design Descriptions. Retrieved from https://ieeexplore.ieee.org/document/5167255 on April 12, 2019.
Doi: 10.1109/IEEESTD.2009.5167255

Other sources:

Microsoft Garcon Project, 2018, October
https://www.youtube.com/watch?v=Ad_EHDcomR8&t=20s

3. Glossary

Term	Definition
CRUD	Abbreviation for Create Read Update Delete operations which are used in Database operations.
Geo-Location API	It is map API provided by Google, allows the user to provide their location to web applications if they so desire.
Instructor	User who is teaching in METU.
IT Staff	Employers who monitoring the use of smart METU Campus.
Logged in User	User who has logged in to Web application or IoT Devices.
METU	Abbreviation for Middle East Technical University.
METU Card frequency	METU Card is the smart card provided by university for students, instructors, staffs. It holds their information via its frequency.
METU Server	METU has own server which has students, instructors, staff information. Smart METU campus project will use it.
MongoDB	MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need.
SATA	Abbreviation for Serial Advanced Technology Attachment. An interface for transferring data between a computer and a storage device.
SSL	Abbreviation for Secure Socket Layer.
Staff	User who is working personnel in METU campus.
Student	User who is studying in METU.

System Developer	They are the group who are developing the Smart METU Campus project.
Unlogged in User	User who has not logged in to Web application or IoT Devices.

Table 1 : Glossary

4. Architectural Views

4.1 Context View

In this view, context of the METU Smart Campus project is defined in the context diagram. In addition, Actor interactions with the system can be found use case Diagram.

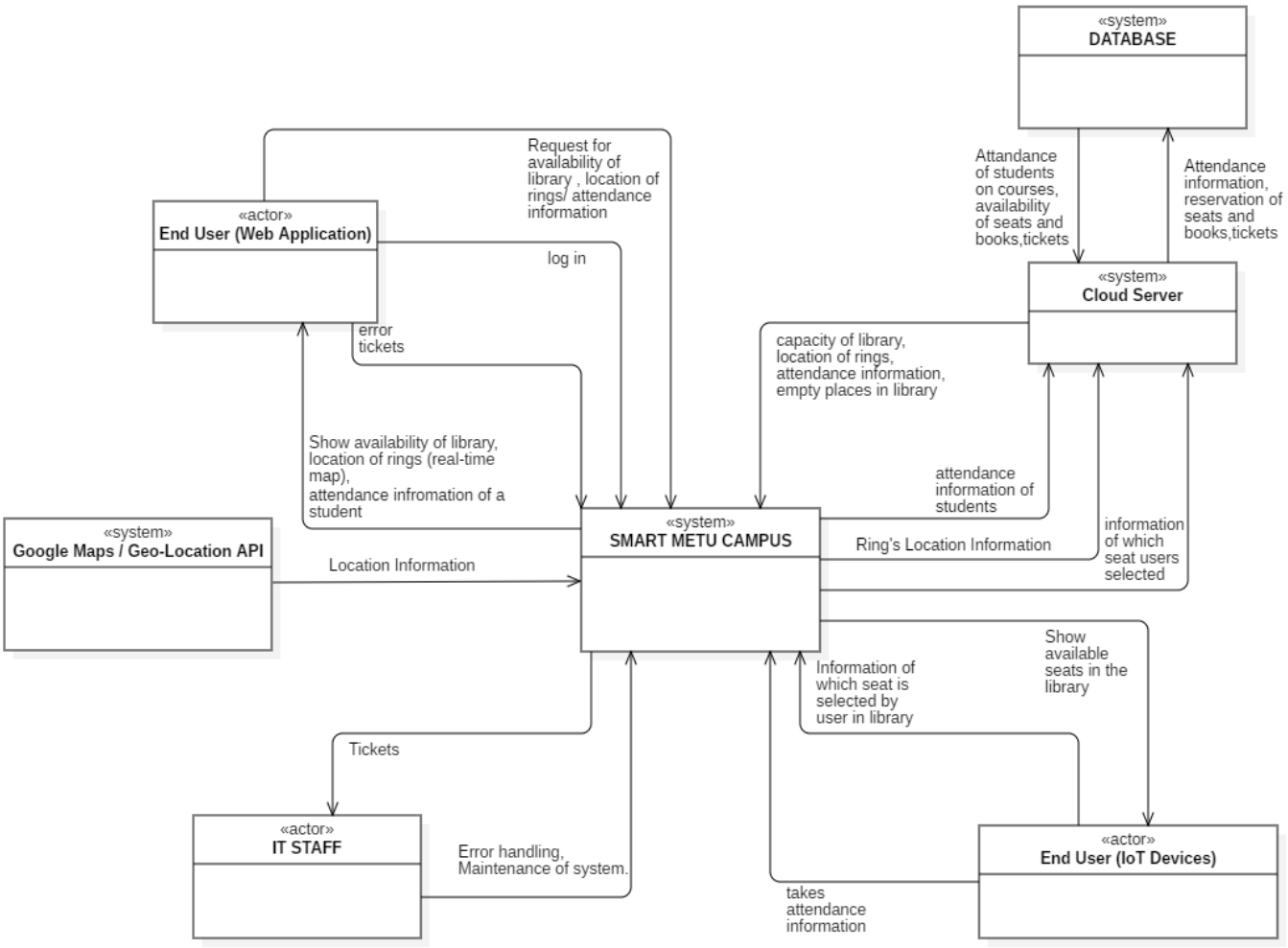

Figure 1: Context Diagram

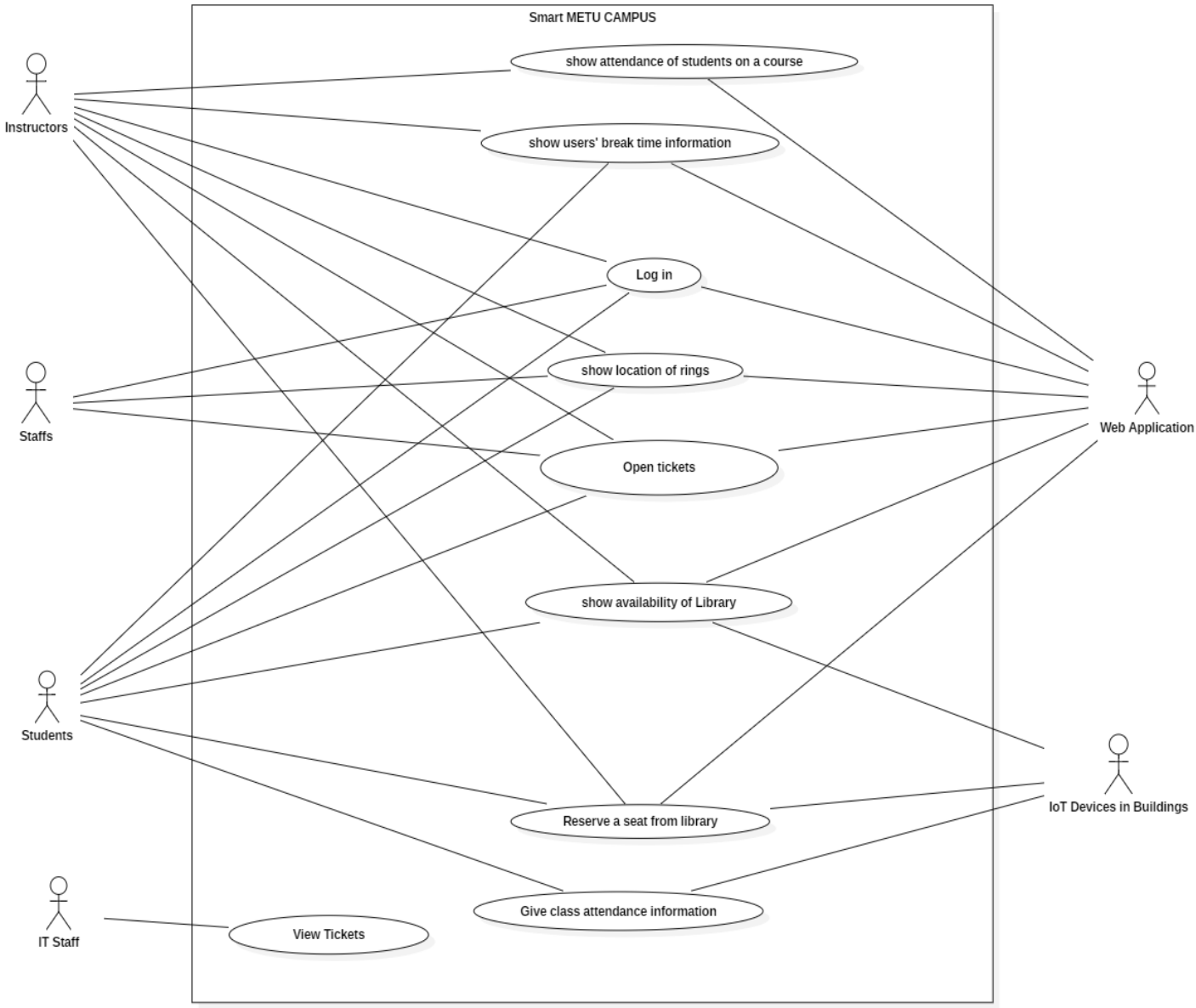

Figure 2 : Use Case Diagram

Use case name	Show location of rings
Actors	Student, Instructors, Staffs (Web Application)
Description	If a user send request for location of rings, according to information from IoT devices, server returns the coordinates to the map (in Web application) with real-time ring monitoring thanks to the Geo-Location API. Besides, it will calculate approximate time that the ring(s) come in to specified in parameters.
Data	Ring color, Station Id
Preconditions	IoT Devices sends their position information.
Stimulus	User requests the data from Web application.
Basic Flow	1 - User requests the location of all rings' location from web application. 2 - Web application send request to Cloud server. 3 - Cloud server fetches the coordinates and sends to Geo Location API. 4 - Geo Location API returns locations to Cloud. 5 - Location data sent back to the Web application. 6 - Web application shows the locations to the user.
Alternative Flow	1 - User requests the data of a ring's location from web application. 2 - Web application send request to Cloud server. 3 - Cloud server fetches the coordinate and sends to Geo Location API. 4 - Geo Location API returns location to Cloud. 5 - Location data sent back to the Web application. 6 - Web application shows the location to the user.

Exception Flow	If Geo-Location API returns an error, it is also written to error log file. If IoT devices don't work (so if data doesn't come), then it is written error log file. In addition, error tickets are sent to the IT staff.
Postconditions	User see the map with position of ring with constantly updated information and time left to reach the station.

Table 2 : Use Case, Show location of rings

Use case name	Show availability of Library
Actors	Student, Instructors, (Web Application, IoT Device)
Description	When a user wants to see available places on the Library, empty places are displayed by pulling the required information from the database.
Data	-
Preconditions	Required user must be either student or instructor.
Stimulus	METU Card is shown to the IoT devices which is in entrance of the library. Or User requested library availability page on web application.
Basic Flow	1-User show him/her card to IoT Device. 2-IoT Device requests available seats from Cloud. 3-Cloud server prepares data and sends to IoT device. 4-IoT Device shows available seats to user on its screen.
Alternative Flow	1-User opened library availability page. 2-Web Application requested available place from Cloud. 3-Cloud server prepares data and sends it to Web application. 4-Web application shows available places to user on corresponding html page.
Exception Flow	If database connection is lost, error message will send to the user. Developers will be informed. In addition, error tickets are sent to the IT staff.
Postconditions	Visualized information of availability of seats of Library will be shown.

Table 3 : Use Case, Show availability of Library

Use case name	Reserve a seat from Library
Actors	Student, Instructors, (Web Application, IoT Device)
Description	After a user views the availability of Library, s/he will be able to select the empty seat to reserve.
Data	User ID, entrance time, seat no
Preconditions	Required user must be either student or instructor and seat must be empty.
Stimulus	METU Card is shown to the IoT devices which is in entrance of the library, then from the IoT device's screen, empty seat will be selected. Or User requested library availability page on web application, then from that page, empty seat will be selected.
Basic Flow	1-IoT Device shows available seats to user on its screen. 2-S/he selects empty seat. 3-IoT Device send selected seat information to cloud along with user info. 3-Cloud takes the information of user and seat reserves seat for the given user and writes it to database. When finished, returns success message sends to IoT device. 4-IoT Device informs user that seat is reserved.
Alternative Flow #1	1-Web application shows available seats to user on library availability page. 2-S/he selects empty seat and send selected seat and user info to Cloud. 3-Cloud takes the information of user and seat and reserves seat for the user and writes it to database. When finished, returns success message sends to Web application. 4-Web application informs user that seat is reserved.

Exception Flow	If database connection is lost, error message will send to the user. Developers will be informed. In addition, error tickets are sent to the IT staff.
Postconditions	Visualized information of availability of seats of Library together with the fullness information of the seat which the user selects the seat will be shown.

Table 4: Use Case, Reserve a seat from Library

Use case name	Give Class Attendance Information
Actors	Student, IoT Device
Description	Students will show their METU ID Card to the IoT Devices which are in the entrance of the classes in beginning and finishing time of course.
Data	METU ID Card's unique info, Class Name, Date
Preconditions	Student must be enrolled to the course.
Stimulus	IoT Device sensors will read the students' METU ID Card.
Basic Flow	1-When the course time start, Students coming to attend class show their ID Card to sensor. 2-IoT Device read the unique information of card and sends to Cloud with class name and date information. 3-Cloud authorizes and writes attendance to the database as unapproved. 4-After class dismissed students must show their ID Card again to complete attendance taking. 5-IoT Device read the unique information of card and sends to Cloud with class name and date information. 6- Cloud server updates students' attendance on database as approved.
Alternative Flow #1	1- Student show their ID Card after class dismissed only. 2- Cloud server does not find starting attendance, writes it as unapproved.
Alternative Flow #2	3- Student show their ID Card before class started only. 4- Cloud server writes it to database as unapproved.

Exception Flow	If database connection is lost, error message will send to the user. Developers will be informed. In addition, error tickets are sent to the IT staff.
Postconditions	-

Table 5: Use Case, Give Class Attendance Information

Use case name	View Tickets
Actors	IT Staff, IT Staff Management Page
Description	If a user or IoT device sent error ticket, IT Staff can get those from Database.
Data	Tickets
Preconditions	-
Stimulus	IT Staff requested tickets.
Basic Flow	1-IT Staff opens IT Staff Management page. 2-Management page requests tickets from Cloud. 3-Cloud fetches tickets from database and sends to the Management page. 4-Management page lists all tickets.
Alternative Flow	-
Exception Flow	-
Postconditions	IT Staff will see all tickets without any exception.

Table 6: Use Case, View Tickets

Use case name	Open Tickets
Actors	User, IoT Device, IT Staff, IT Staff Management Page
Description	When a user face with a problem, s/he can send tickets about the situation to the IT Staff via Web application open ticket page.
Data	User info, Content of Ticket, Date
Preconditions	-
Stimulus	User open a ticket from web application.
Basic Flow	1-User request open ticket page and enters ticket information. 2-Web application sends it to the Cloud. 3-Cloud writes it to the Database and return success message. 4-User sees the success message.
Alternative Flow	-
Exception Flow	If internet or database connection is lost, error message will send to the user. Developers will be informed. In addition, error tickets are sent to the IT staff.
Postconditions	User see a successful or failure messages.

Table 7: Use Case, Open Tickets

Use case name	Show Attendance of Students on a Course
Actors	Instructors, Web Application
Description	Instructor will be able to fetch all attendance data from server, so he/she can evaluate students' attendance.
Data	Course Code, Instructor ID
Preconditions	Course must be given in that semester by the instructor
Stimulus	Instructor opens corresponding page on Web application.
Basic Flow	1- Instructor opens show course attendance page. 2- Web application requests attendance information from Cloud. 3- Cloud server fetches data from database and sends back to web application. 4- Web application shows information to instructor.
Alternative Flow	-
Exception Flow	If database connection is lost, error message will send to the user. Developers will be informed. In addition, error tickets are sent to the IT staff.
Postconditions	All attendance data of given course will be listed without any exception Including unapproved attendance.

Table 8: Use Case, Show Attendance of Students on a Course

Use case name	Show User's Break Time Information
Actors	Student, Instructors, Web Application
Description	When a user who holds a desk in library takes a break, there starts a countdown (30 mins) and user can see how many minutes they got.
Data	User ID
Preconditions	A seat must be reserved by the user in the library.
Stimulus	User opens library reservation page.
Basic Flow	1-User opens library reservation page on web application. 2-Web application requests the data from Cloud. 3-Cloud fetches break time from database and calculates timestamp, which is returned to web application. 4-Web application show the time passed till break time.
Alternative Flow	1-User opens library reservation page on web application. 2-Web application requests the data from Cloud. 3-Cloud fetches break time from database and calculates timestamp, timestamp is over limitation, which implies that seat is no longer reserved for user. Server returns a message stating that seat is freed from his/her charge. 4-Web application shows message.
Exception Flow	If database connection is lost, error message will send to the user. Developers will be informed. In addition, error tickets are sent to the IT staff.
Postconditions	User must know how much time is passed, if timestamp is beyond limitation user must be informed about it.

Table 9: Use Case, Show User's Break Time Information

Use case name	Login
Actors	User, Web Application
Description	Users are needed to authorized to use utilities provided by the system.
Data	User Credentials. Or METU ID Card unique information.
Preconditions	-
Stimulus	Users shows their card to IoT Devices. Users login to web application with their credentials.
Basic Flow	1-User shows their card to IoT Devices. 2-IoT Device sends unique information of card to Cloud. 3- Cloud sends data to METU Servers to authorize user. 4-METU Server authorize user and return ID of user to Cloud. 5-Cloud send authorization information to IoT Device and writes ID to cache created for short-time for that session. 6-User authorized.
Alternative Flow #1	1-User shows their card to IoT Devices. 2-IoT Device sends unique information of card to Cloud. 3- Cloud sends data to METU Servers to authorize user. 4-METU Servers does not authorize user and returns an error message. 5-Cloud sends message to IoT Device.
Alternative Flow#2	1-User enters credentials on Web application. 2-Web application encrypts them and sends to Cloud. 3-Cloud sends data to METU Servers to authorize user.

	4-METU Server authorize user and return ID of user to Cloud. 5-Cloud send authorization information to Web application and writes ID to cache created for short-time for that session. 6-User authorized.
Alternative Flow#3	1-User enters credentials on Web application. 2-Web application encrypts them and sends to Cloud. 3-Cloud sends data to METU Servers to authorize user. 4-METU Servers does not authorize user and returns an error message. 5-Cloud server returns the message to Web application.
Exception Flow	If database connection is lost, error message will send to the user. Developers will be informed. In addition, error tickets are sent to the IT staff.
Postconditions	-

Table 10: Use Case, Login

4.2 Composition View

In this view, Smart METU Campus system's components, external components and their functionalities are described via component diagram and deployment diagram. Detailed explanation of them can be found in the following content.

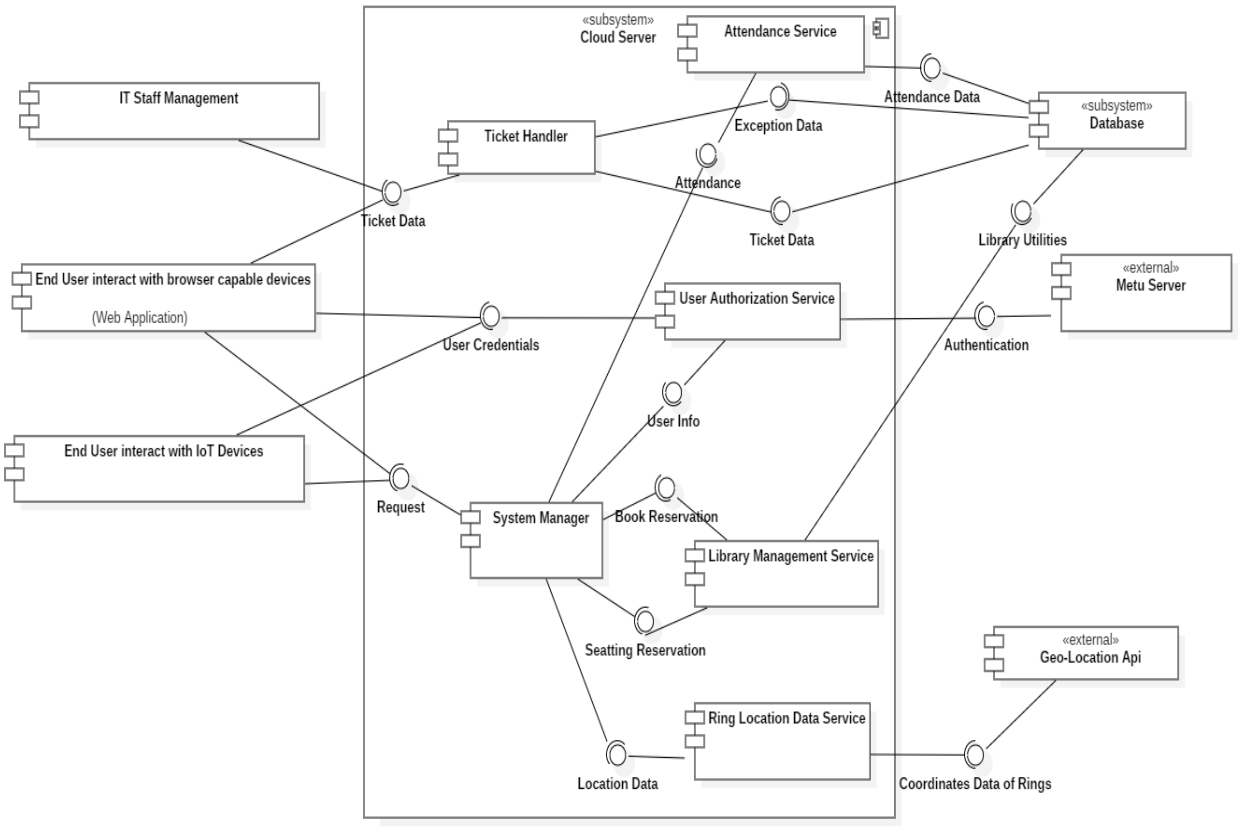

Figure 3: Component Diagram

Design Rationale:

- Cloud server subsystem consist of different components communicating with the external systems, mostly with METU Servers, in addition, it uses Geo-Location API Service from Google for ring locations service.
- System manager is the main component that handles the requests coming from client with services. It also authenticates users in every request to prevent system vulnerability.
- Ticket handler is the service is responsible for that tickets opened from users to be delivered to IT Staffs through IT Staff Management client.
- User Authorization Service is service that gets either user credentials or ID Card information (frequency, etc.) and sends them to METU Servers to authenticate users. If authentication fails, it informs System manager and system manager cancel request of user.
- Attendance Service is the service for student attendance-tracking. All data coming through IoT devices located in entrance of classes, is being handled by this service and saved to the database. Besides, if students or instructors requires to see the past-attendance information, it prepares the data and sends back.
- Library Management Service is the service for seating reservation and the book reservation in the library. Request sent when users reserves a seat from IoT devices in the entrance of library, or reserves books from either the IoT devices or Web application is being handled by this service.
- Ring Location Data Service is the service responsible for providing instant location of rings and calculation of arrival time by communicating with Geo-Location API to determine exact location on a map.
- IT Staff Management is a client-side web page. By the help of this web page, IT Staff can see the tickets and handles them if necessary.
- Web Application is the main client that users can communicate with the cloud server. They can see attendance information, library utilities availability, ring locations with the help of web application.
- IoT Devices have limited interface. Students can give attendance with the help of IoT devices in the entrance of classes. Users can benefit the library utilities with the help of IoT devices in the entrance of library.

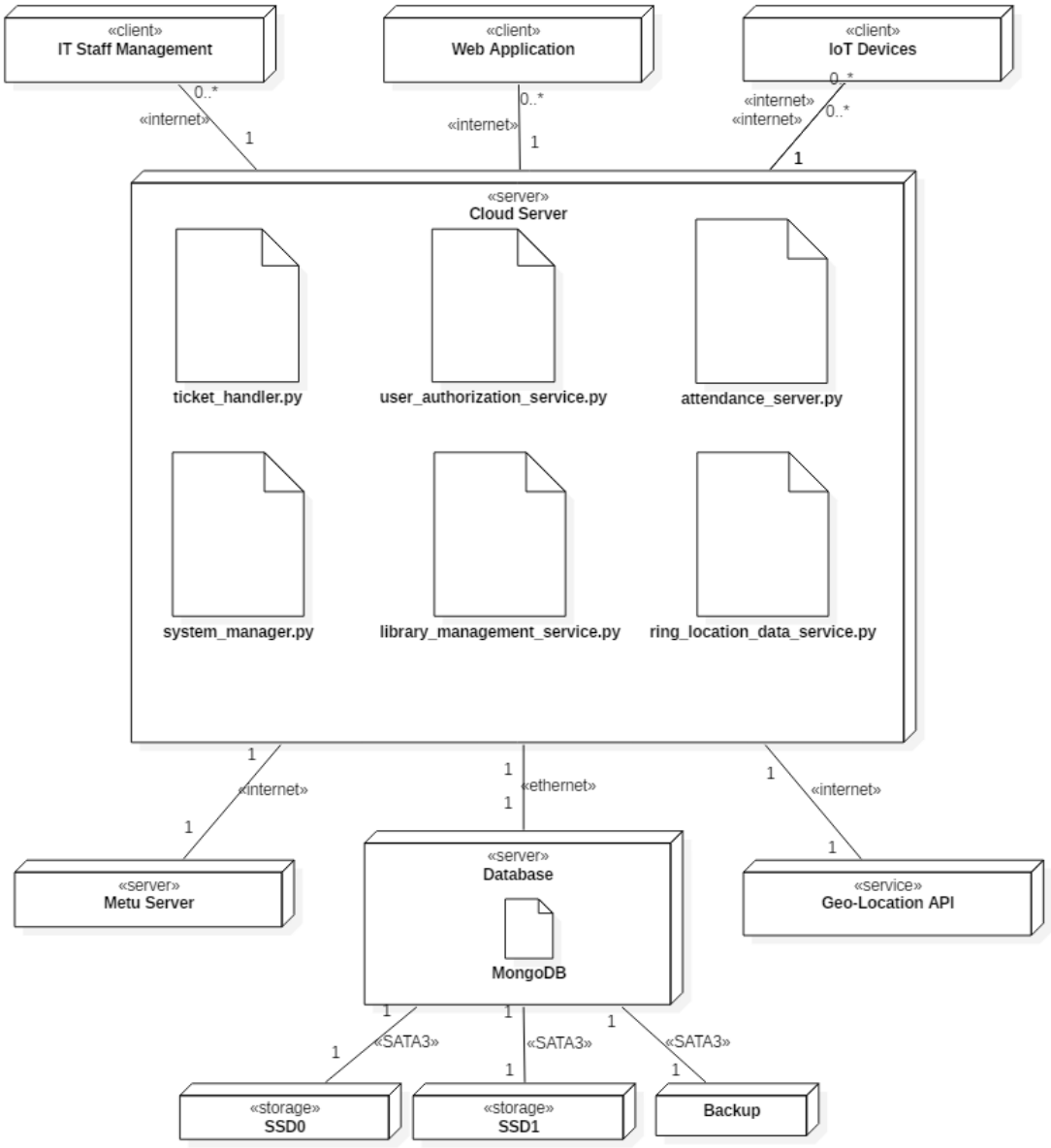

Figure 4:Deployment Diagram

Design Rationale:

- We decided to use Python and MongoDB in implementation of services for maintaining and easy coding issues.
- End users communicates with the cloud servers with the help of two clients, Web application and IoT devices.
- We have one SSD for storing library data, one SSD for attendance data. We used SATA3 because it provides more storage capacity. We also used a Backup storage in case of any system failure, which will be backed up weekly.

- Only Geo-Location API and clients uses internet to communicate with cloud server. Subsystems and database will have wired connections to escape the internet's vulnerability, latency security issues.

Such encrypted communication protocols as SSL certificate, HTTPS will be used in internet-based connections.

4.3 Information View

In this view, interfaces of the system and system, database relations will be explained with System Class Diagram, Database Class diagram and their further explanations will be appeared in Operation Description, Design tables and design rationales.

4.3.1 Interfaces

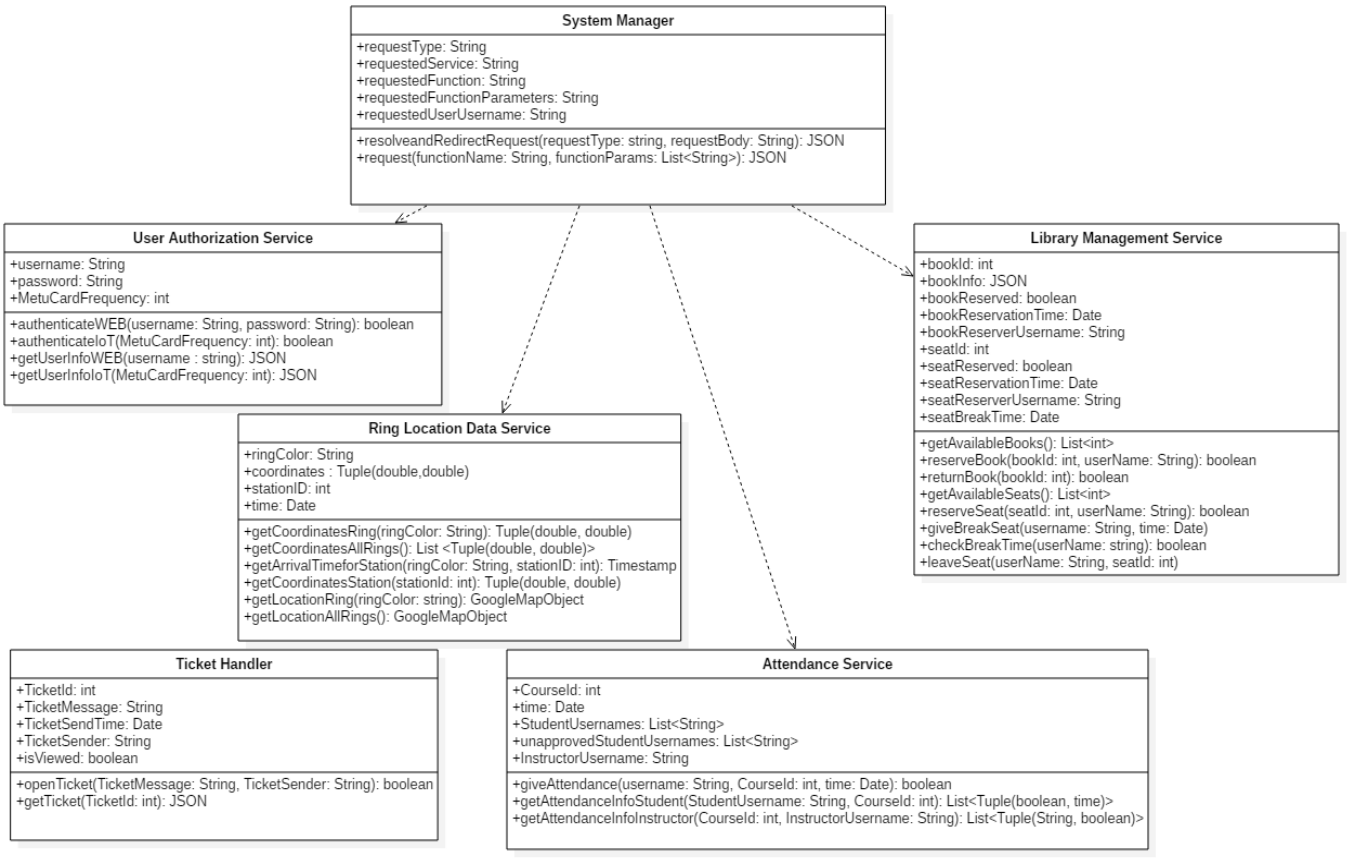

Figure 5: Class Diagram

Operation	Description
request	Users request on Web application and IoT Devices will be sent via this operation. Requested function and corresponding parameters will be encrypted to a JSON object.
resolveandRedirectRequest	JSON data will include requestedService, requestedFunction, requestedFunction's parameters and requestedUser. Operation will resolve them and will redirect to corresponding service function with proper parameters.
authenticateWEB	Users credentials(basically username and password) will be sent to User Authorization Service to authorize user. Service will communicate with METU Server end will authorize user.
authenticateIoT	METU ID Card's frequency will be sent to User Authorization Service to authorize user. Service will communicate with METU Server end will authorize user.
getUserInfoWEB	When a user requests any service requires user info from web application, System Manager will fetch info with this operation that info.
getUserInfoIoT	When a user requests any service requires user info from IoT Device, System Manager will fetch info with this operation that info.
getCoordinatesRing	Ring Location Data Service will fetch coordinate of given ring from gps services with this operation.
getCoordinatesStation	Ring Location Data Service will fetch coordinate of given Station from gps services with this operation.
getArrivalTimetoStation	Estimated time of arrival for given ring will be calculated according to fetched coordinate information via getCoordinatesRing and Station information fetched via getCoordinatesStation.
getLocationofRing	Given ring's location is requested via this operation. Service will fetch location from Geo-Location api and send the map object as response.
getLocationofAllRings	All rings' location is requested via this operation. Service will fetch location info from Geo-Location api and send the map objects as response.

openTicket	Users will be able to open ticket about errors & failures or any questions about
getTicket	IT Staff will get ticket info with this operation.
giveAttendance	Every time students show their ID Card to IoT Devices in entrance of classes, this function will be triggered to get attendance.
getAttendanceInfoStudent	Students will get their attendance info for given course.
getAttendanceInfoInstructor	Instructors will get attendance info of students of course he/she is giving.
getAvailableBooks	Returns all available books in library.
reserveBook	Users will reserve given book for a period.
returnBook	Users will give up given book that they already reserved.
getAvailableSeats	Users will get list of available seats.
reserveSeat	Users will reserve given seat in library for themselves.
giveBreakSeat	Users informs system that they will be apart for a while.
checkBreakTime	Users can check time elapsed since they give break.
leaveSeat	Users will inform system that they will no longer use given seat.

Table 11 : Operation Descriptions

Operation	Inputs	Outputs	Exceptions
request	FunctionName, FunctionParameters	Response of function as JSON Object	Database Server is not available
resolveandRedirectRequest	requestType, requestBody	Response of redirected function as JSON Object	Database Server is not available
authenticateWEB	Username, password	isAuthenticated OK or NOT OK	Database Server is not available
authenticateIoT	METU Card Frequency	isAuthenticated OK or NOT OK	Database Server is not available
getUserInfoWEB	username	JSON Object representing user	Database Server is not available
getUserInfoIoT	METU Card Frequency	JSON Object representing user	Database Server is not available
getCoordinatesRing	ringColor	Coordinates as tuple	Database Server is not available
getCoordinatesStation	StationID	Coordinates as tuple	Database Server is not available
getArrivalTimetoStation	ringColor, StationID	Timestamp	Database Server is not available
getLocationRing	ringColor	Google Map Object	Database Server is not available
getLocationAllRings	-	List of Google Map Objects	Database Server is not available
openTicket	TicketMessage, TicketSender	isOpened OK or NOT OK	Database Server is not available

getTicket	TicketID	JSON Object representing Ticket	Database Server is not available
giveAttendance	StudentUsername, CourseID, time	isAttendanceTaken OK or NOT OK	Database Server is not available
getAttendanceInfoStudent	StudentUsername, CourseID	List of classes with a boolean representing if the student attended class or not.	Database Server is not available, Student is not enrolled to given Course
getAttendanceInfoInstructor	InstructorUsername, CourseID	List of classes with a boolean representing if students attended class or not.	Database Server is not available, Instructor is not giving given Course
getAvailableBooks	-	List of available books	Database Server is not available
reserveBook	bookID, Username	isReserved OK or NOT OK	Database Server is not available
returnBook	bookID	isReturned OK or NOT OK	Database Server is not available
getAvailableSeats	-	List of available seats	Database Server is not available
reserveSeat	seatID, Username	isReserved OK or NOT OK	Database Server is not available, user is not student or instructor
giveBreakSeat	Username, time	-	Database Server is not available

checkBreakTime	Username	Timestamp	Database Server is not available
leaveSeat	Username, seatID	-	Database Server is not available

Table 12 : Operation Design

Design Rationale:

- System Manager is responsible for handling the communication of End user with Services. It takes requests, resolves it and make appropriate service function calls.
- Authentication is virtually handled by Authentication Handler, but it is actually done in METU Servers. Handler does communication with METU Server and authenticates and/or get user info. Web application and IoT Device have different authentication methods, so for both there exist different get & authenticate functions.
- Ring Location Service is responsible for ring tracking. It has operations for both single ring and all rings.
- Library Management Service is responsible for both seat reservation and book reservation. Users must see available seat and books before reserving, so there exists for getting available books or seats. Besides, to reserve books and seat there exists operations which interacts with Database. Also break time tracking is done with corresponding operations.
- When user request something, first it authenticate itself, and then sends requests to System Manager.
- When user sees a problem, he/she opens a ticket via Ticket Handler so that IT Staff can handle error.

4.3.2 Database Operations

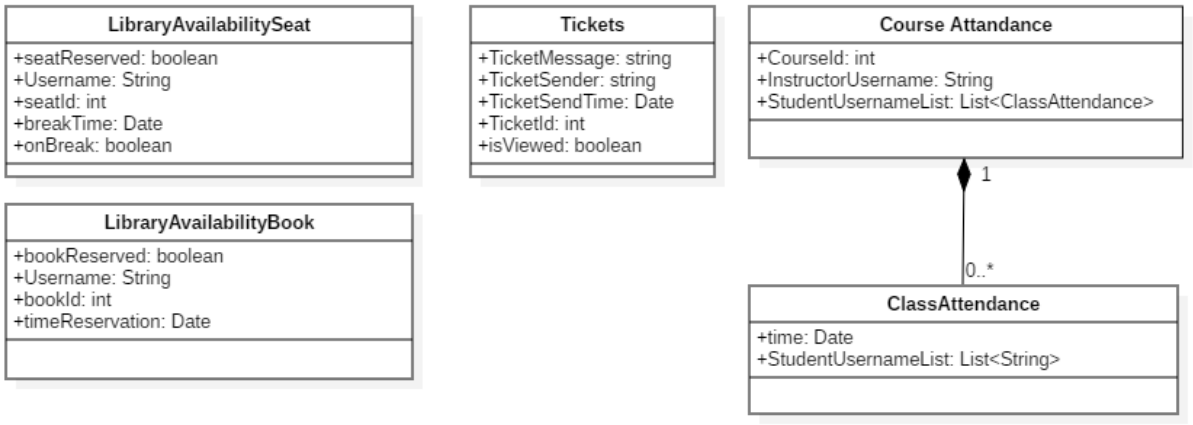

Figure 6 : Database Class Diagram

Operation	CRUD Operations
openTicket	Create: Ticket Read: Update: Delete:
getTicket	Create: Read: Ticket Update: Delete:
giveAttendance	Create: ClassAttendance Read: Update: ClassAttendance, Course Attendance Delete:
getAttendanceInfoStudent	Create: Read: ClassAttendance, Course Attendance Update: Delete:
getAttendanceInfoInstructor	Create: Read: ClassAttendance, Course Attendance Update: Delete:

getAvailableBooks	Create: Read: LibraryAvailabilityBook Update: Delete:
reserveBook	Create: Read: LibraryAvailabilityBook Update: LibraryAvailabilityBook Delete:
returnBook	Create: Read: LibraryAvailabilityBook Update: LibraryAvailabilityBook Delete:
getAvailableSeats	Create: Read: LibraryAvailabilitySeat Update: Delete:
reserveSeat	Create: Read: LibraryAvailabilitySeat Update: LibraryAvailabilitySeat Delete:
giveBreakSeat	Create: Read: LibraryAvailabilitySeat Update: LibraryAvailabilitySeat Delete:
checkBreakTime	Create: Read: LibraryAvailabilitySeat Update: Delete:
leaveSeat	Create: Read: Update: LibraryAvailabilitySeat Delete:

Table 13 : CRUD Operations

Design Rationale:

- For library utilities availability, basically seats and books, there exists separate collections. Each seat and book have always a document on these documents and every reservation and leave operations updates existing documents.

- Tickets also stored in separate collection. When opened a ticket by either user or an exception caused by database, it is stored on database. So, IT Staff can get tickets and review it to handle errors.

- Course attendance are hold in two different collections, Course Attendance and Class Attendance. Course Attendance stores Course ID and Instructor giving course so validation may be done quickly, and holds a list of reference to Class Attendance documents. A Class Attendance document holds time information and approved or unapproved student list.

4.4 Interface View

In this view, the internal interfaces, the external interfaces will be explained and some use case flow is described via sequence diagram.

4.4.1 Internal Interfaces

The interface between the System Manager and End User (with IoT Devices and Web Application):

The system manager handles all requests coming from end user via IoT or Web Application. Web Application sends it's request in a similar way querying database, sends string of requested function name and list of parameters to that function. System Manager resolves request and redirect to appropriate service with appropriate function call. It is same as when IoT Device sends requests, a string is prepared and sent to System Manager and system manager redirects request.

Design Rationale:

It makes clear developing on client side (Web application and IoT Device) to send request to cloud server. And makes it not affected from changes internally done. Also, validation of requests is handled properly, if invalid request is given, it handles it securely without any harm in first place.

The interface between the System Manager and the Ring Location Data Service:

When system manager is requested with ring location, it redirects it to Ring Location Data Service with validated parameters. This interface provides system manager to get map which shows ring location on it.

Design Rationale:

Users cannot access ring location data service directly due to security issues. Instead, they sent their request to system manager asking ring location, system manager serves as secure bridge between end user and ring location service and redirects it through this interface.

The interface between the System Manager and the Library Management Service:

Library Management Service is the service that handles of reservation of books and seats in the library. Users either on IoT Devices in the library or on Web Application may request available books and seats or make reservation. But, these requests cannot be directly sent to Library Management Service. Instead, it passes to the system manager and after validation, redirected to Library Management Service via this interface.

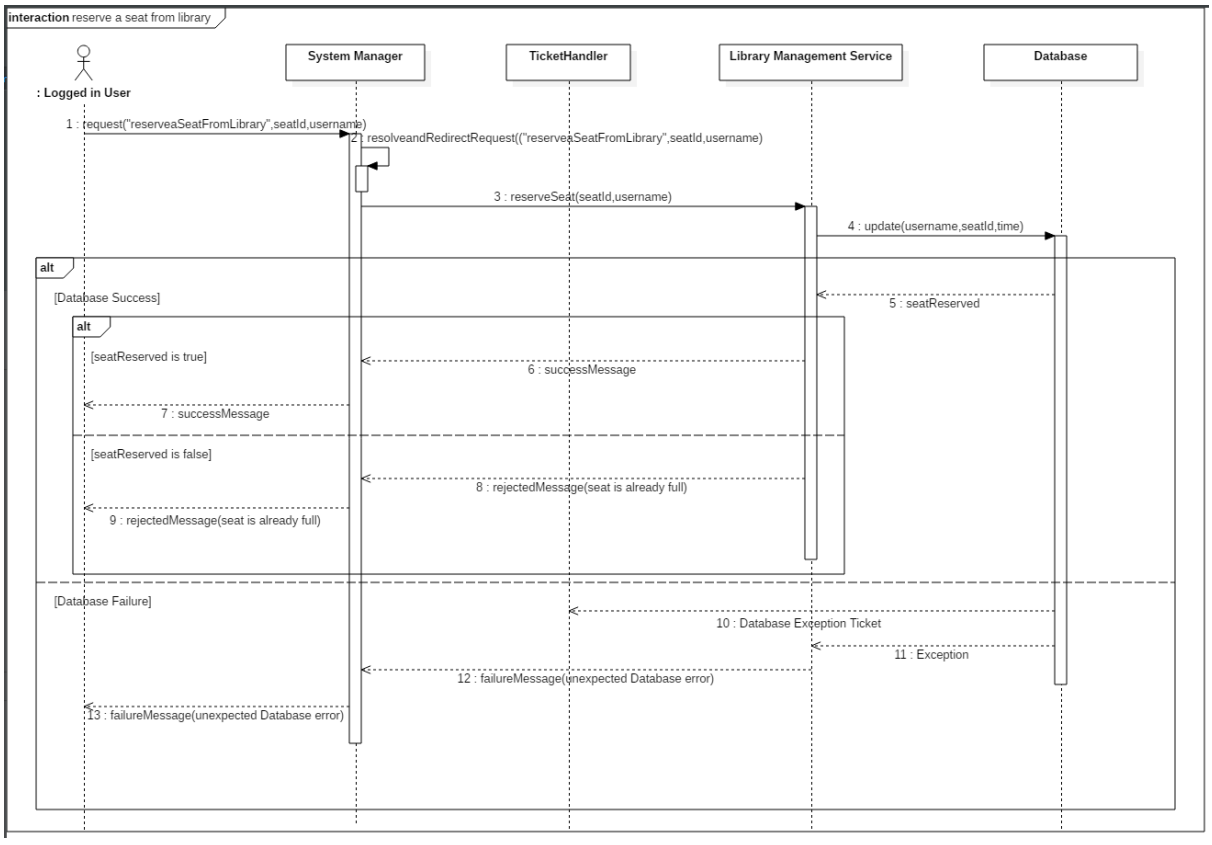

Figure 7 :Reserve a seat from library sequence diagram showing the interface between System Manager and Library Management Service components of the system

Design Rationale:

Users cannot access Library Management service directly due to security issues. Instead, they sent their request to system manager about their requests, system manager serves as secure bridge between end user and Library Management Service and redirects it through this interface.

The interface between the System Manager and the User Authentication Service:

Some Services may require user info. Since, info is not stored in database, it must be fetched from METU Server. Service communicates with METU Server is User Authentication Service. As a result, when any service requires user info, system manager fetch user info via User Authentication Service with this interface.

Design Rationale:

Other than System Manager, any service has no access to User Authentication Service, thus cannot get user info by themselves. Instead, system manager will handle this and send to the service as parameter. Besides, it makes project compact. For example, when a new service requiring user info is added, instead of creating a whole interface, it will be handled during service call from system manager. Which makes development and maintenance is easy.

The interface between the System Manager and the Attendance Service:

Attendance Service is the service which handles students' attendance to the classes and instructors giving the course. When students show their ID Cards to IoT Devices in the entrance of classes, an attendance taking request is sent to System Manager. System Manager resolves it and sends to Attendance Service via this interface. Also, any other requests like checking previous attendance information will be going through this interface.

Design Rationale:

Users cannot access Attendance Service directly due to security issues. Instead, they sent their request to system manager about their requests, system manager serves as secure bridge between end user and Attendance Service and redirects it through this interface.

The interface between the IT Staff Management, End User and the Ticket Handler:

Ticket Handler is different from other services. Ticket Handler is accessed directly by user and IT Staff Management. This interface provides a communication bridge between IT Staff and End User.

Design Rationale:

Ticket Handler is independent from System Manager due to it is an error handling process. It provides direct communication between IT Staff and End User. So, cloud server development area is not affected any change done on Ticket Handler Service.

The interface between the Ticket Handler and the Database:

This is the basic interface provides that ticket handler stores opened tickets and sends them to IT Staff when requested. It provides write & update & read operations on Database.

Design Rationale:

This is a basic API that providing simple function calls to get and post to database. It makes the Ticket Handler to query-language free. Therefore, any differentiation on database system will not affect Ticket handler.

The interface between the Library Management Service and the Database:

This is the basic interface provides that Library Management stores and operates on books & seats info on database. It provides Library Management Service write & update & read operations on Database.

Design Rationale:

This is a basic API that providing simple function calls to get and post to database. It makes the Library Management Service to query-language free. Therefore, any differentiation on database system will not affect Library Management Service.

The interface between the User Authentication Service and End User (with IoT Devices and Web Application):

Every time user requests something from System Manager, it must have an auth-key taken from METU Server that indicates he/she is a member of METU, so system manager proceed that request. User Authentication Service provides an interface to user that gets auth key from METU Server corresponds to given credentials (Web Application) or METU ID Card frequency (IoT Device). Auth key checked internally on System Manager by using interface between System Manager and User Authentication Service.

Design Rationale:

Basic log in service. Since user data is fetched from METU Servers, and an user may login once and make several requests, authentication may be separated from System Manager to reduce overhead.

The interface between the Database and the Attendance Service:

This is the basic interface provides that Attendance Service stores and operates on attendance info of students on classes on database. It provides Attendance Service write & update & read operations on Database.

Design Rationale:

This is a basic API that providing simple function calls to get and post to database. It makes the Attendance Service to query-language free. Therefore, any differentiation on database system will not affect Attendance Service.

4.4.2 External Interfaces

4.4.2.1 User Interfaces

Student Interface: Students are allowed to reserve books and seats from library, view ring location and give attendance to classes they attended. Whereas in web application there exists pages for all functionality that a student can do except giving attendance, in IoT Devices, seat reservation may be done with the help of a screen, but giving attendance is done with showing ID Card to IoT Classes in entrance of classes.

Instructor Interface: Instructors are allowed to reserve books and seats and view ring location as well as students, but they differ from students in attendance. They have permission to collect attendance information and check coming students his/her classes. Whole functionality is available on web application. In addition to that, instructor may use IoT Devices in the library to reserve seats.

Staff Interface: Staffs are the working personnels in the campus. They have access to ring locations, and book reservations. Other functionalities are not provided for them. These are available as web pages on web application, they do not have an interface in IoT devices.

IT Staff Interface: IT Staffs are permitted to get opened tickets by end users. They have basic web interface to view tickets and work on problems.

4.4.2.2 System Interfaces

The Interface between the User Authentication Service and METU Server:

User Authentication authenticates users by inputted credentials or ID Card frequencies read by IoT Devices, by sending them to METU Server. This authentication, actually, done on METU Server. So, this interface provides only credential/frequency sending, and authenticated/not-authenticated response getting. If user authenticated, User Authentication Service create an authentication key and sends to the user.

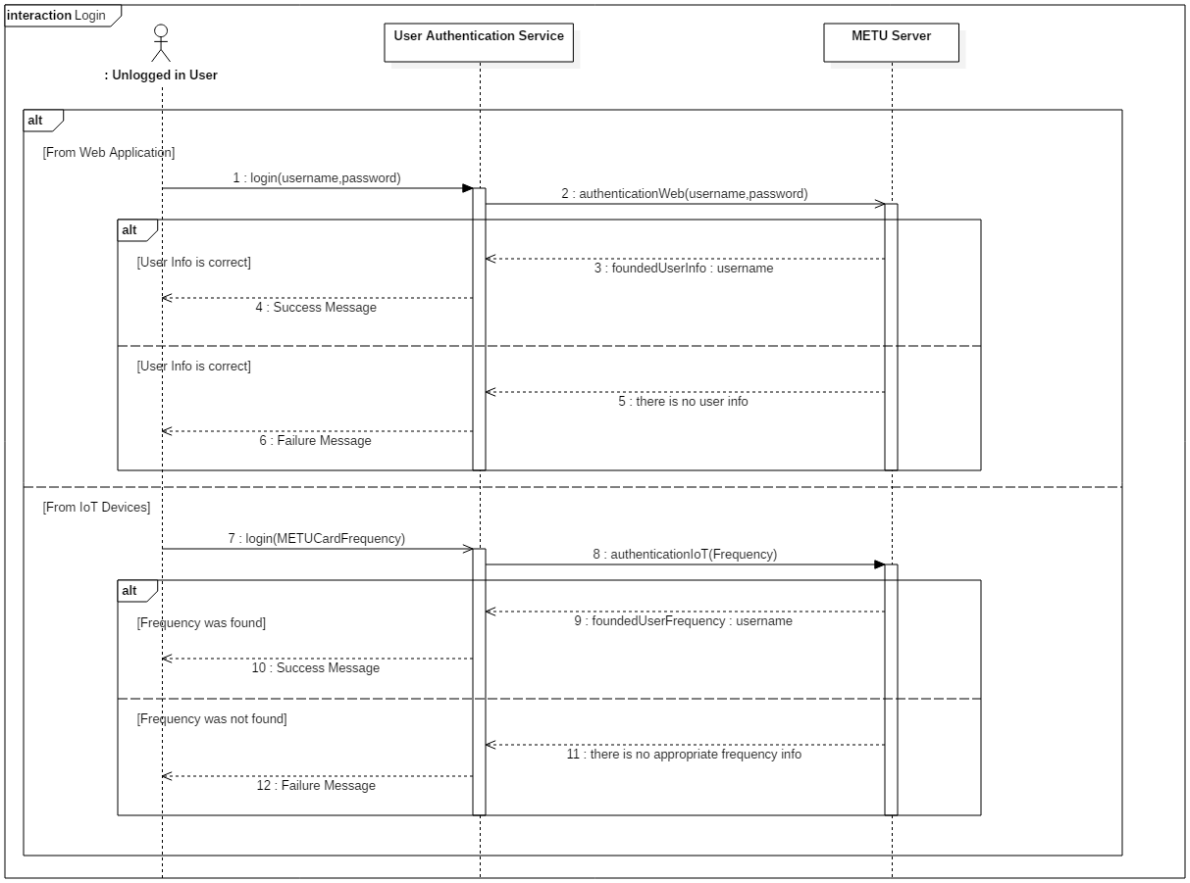

Figure 8 : Log in sequence diagram showing the interfaces between User Authentication Service and external component METU Server

Design Rationale:

Since no critical user data is used in the system, credentials pass through User Authentication to METU Server and if authenticated, creates and temporary authentication key maps user to his/her account on Smart METU Campus System. So, both security by not storing critical data, and authentication issues are handled in a single service. Which concludes that it is adequate that this interface provides communication between User Authentication Server with METU Server for only credential/Card frequency and is authenticated information.

The Interface between the Ring Location Data Service and the Geo-Location API:

Ring Location Data Service is responsible for creating maps showing instant location of rings. The service gets map Object from Geo-Location API by sending it coordinates of rings.

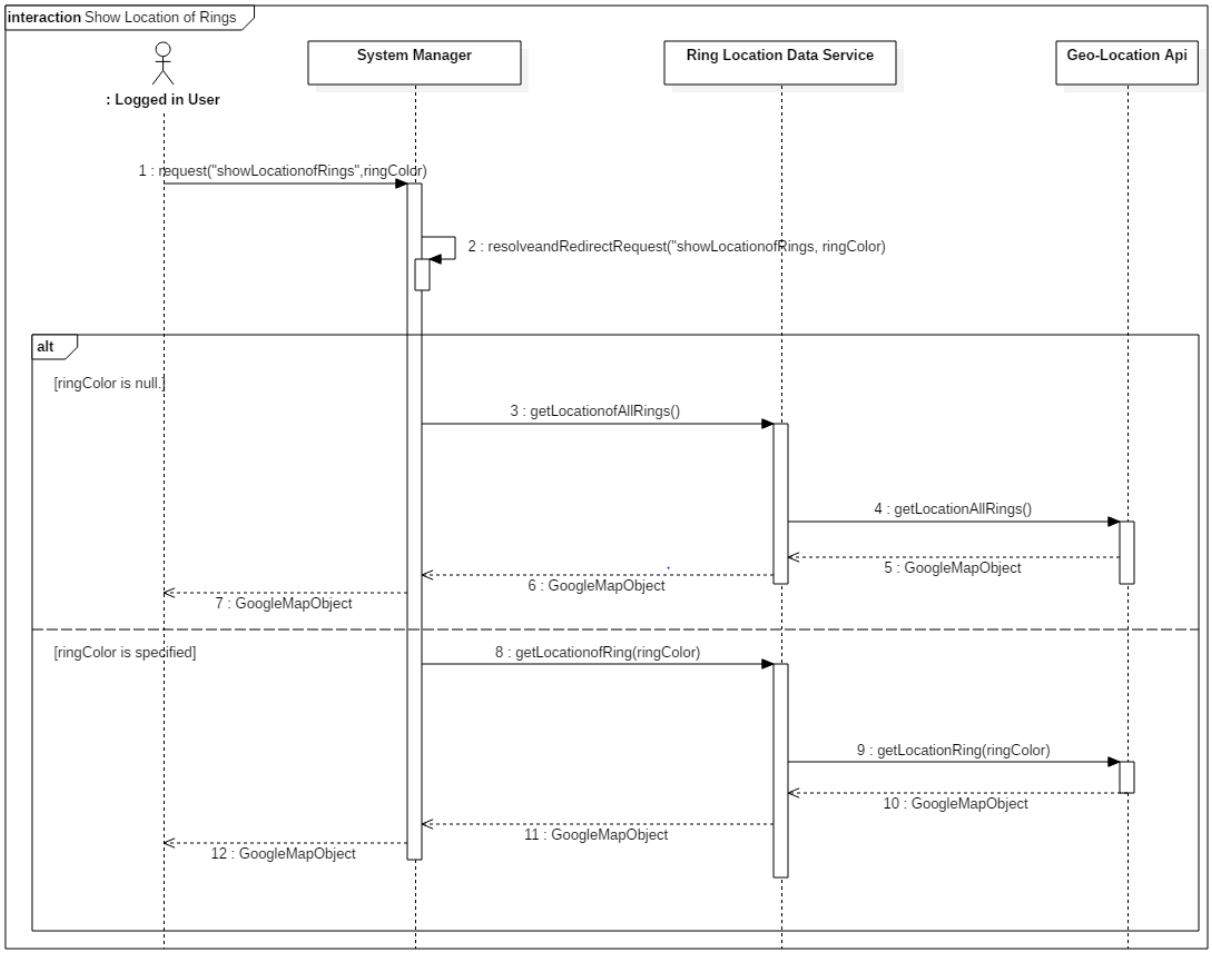

Figure 9 : Show location of rings sequence diagram showing the interfaces between Ring Location Data Service and external component Geo-Location API

Design Rationale:

Ring Location Data Service needs mapobjects created by Geo-Location API. This interface provides the service to send coordinates to API, and in return get mapobjects showing instant locations. Once mapobject is fetched, connection is not closed, and by the help of a web socket, coordinates are being continued to send to API and instant location is updated on mapobject.