Software Design Description

Garcon

Version 2.0

METU CENG350 2018-2019 Spring

Orçun BAŞŞİMŞEK

2098804

Record of Changes

Date	Version	Description of Change
15.04.2019	1.0	First release. Component and Deployment diagrams.
26.04.2019	2.0	Final release. Full SDD document

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose of the System

"Garcon" system basically relies on "Smart Campus" idea. The system is developed for providing various facilities about basic campus related activities such as reporting a campus related issues, asking information about buildings or activities at the campus, booking transactions etc. Thanks to the large number of IoT devices placed in various parts and buildings of the campus, it is aimed that all campus people (students, academic people, staff, guests etc.) access and utilize these services in an easy way.

1.2 Scope

- The system will consist of mainly a lot of IoT devices called "Garcon" when viewed by normal campus people from outside. Main goal of these devices, as the name "Garcon" implies, are providing campus related services to people to facilitate their life at the campus.
- Garcon devices will be basically voice-activated and very intelligent devices. They just need to ID card verification to be ready for help to the users.
- By swiping their ID card, campus people can communicate with Garcon devices with just voice, and can ask some directions, campus activities, accommodation or transportation opportunities in the campus or they can make reservations at restaurants and guesthouses in the campus or they can report any infrastructural or cleaning related issues to be solved quickly by related staff.
- The system will have a voice-to-text translator device inside Garcon IoT devices and will translate human voice commands to text for understanding user intent at the back-end services.
- At the back-end, the system will contain Azure IoT Cloud Service that is developed by Microsoft. This Cloud service will be responsible for understanding the meaning of user intent and deciding and executing the tasks as a response to the user.
- The system will work in collaboration with CRM (Customer Relationship Management) System. IoT devices can create tickets according to user's reports or booking activities, and CRM System will be responsible for fulfilling user requests according to these tickets.
- There will be a sensor system as hardware that is connected with Azure IoT Cloud Service continuously, and Garcon devices can gather locational, infrastructural information from these sensors every time needed (e.g. when creating a ticket about heating problem in some classroom).
- There will be a University Information service for giving response to the user about basic campus related information such as map, ring hours, cafeteria menus etc.

- The system will also work in collaboration with Microsoft Campus Link Application. This application will be accessed from mobile devices and users can reach additional information about their requests from Garcon devices and they can also trace their booking requests or tickets' status.

1.3 Stakeholders and their concerns

The stakeholders of the Garcon system can be divided into 4 different categories. Commonly, all of these stakeholders of the system need to user-friendly and simple UI for using the system and communicate with the system by just voice.

Campus People: They are ordinary Campus people such as students or academic people in the campus, and they are expected to benefit most from the system. Their main concern is to get clear and accurate response from the Garcon according to their needs or requests. Moreover, communication quality of Garcon (e.g. voice power, voice clarity, response time) and Garcon's concurrency with Microsoft Campus Link application are important for them. They also need to trace their requests' status or results (e.g. result of booking a room in guesthouse request) from Microsoft Campus Link application precisely.

Guests: Guests are people who come to campus from outside for a specific reason such as a conference, university festival or visiting some campus people. They will use the system by their guest card that they received from security while entering to the campus. Their primary concern is to get university and campus related information from the Garcon according to their specific visit reasons.

Campus Staff: They also utilize from the system like ordinary Campus people, and their concerns are same with ordinary Campus people from this point of view. However, they also need to effective and easy-to-use staff UI (user interface) while using Garcon devices because they need to be able to inform the system about solving issue tickets.

System Admins: They are special personnel whose jobs are analyzing and solving problems about the system and maintaining the system basically. They firstly need to be identified as System admin by Garcon according to their special ID numbers. Then, their main concern is to get expected behavior from the Garcon while they communicate with Garcon by special administrative commands to do their jobs.

2 References

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

IEEE. (2009, July 20). 1016-2009 – IEEE Standard for Information Technology — Systems Design — Software Design Descriptions.

Other Sources:

Microsoft's Smart Campus IoT and AI project "Garcon". (2018, October 30). Retrieved March 17, 2019, from https://channel9.msdn.com/Shows/Internet-of-Things-Show/Microsofts-smart-campus-IoT-and-AI-project-Garcon

Azure IoT. (n.d.). Retrieved March 17, 2019, from https://azure.microsoft.com/en-us/overview/iot/

Microsoft CampusLink Application. (2018, December 19). Retrieved March 17, 2019, from https://appgrooves.com/android/com.refmobile/microsoft-campuslink/microsoft-corporation

Shankaran, Ganesh. *Garcon Architectural View*.

3 Glossary

Term	Definition
ID card	Identification card. The entity that is used for user's authentication and authorization transactions.
CRM System	Customer Relationship Management System. System component that enables the conclusion of booking and issue ticket transactions.
API	Application Programming Interface
IoT device	Internet of things device. Garcon devices that are placed on different points of campus.
UI	User Interface
Campus people	Students, academic people. The target users and stakeholders of the system.
WSN	Wireless Sensor Network. The network that is used for communication between sensors and sensor service.
SQL	Structured Query Language
DBMS	Database Management System
SATA	Serial Advanced Technology Attachment. The bus interface used for storage.
CRUD	Create, read, update, delete operations on database.

Table 1: Glossary

4 Architectural Views

4.1 Context View

Context of the system are explained in a general way in this view. In the context diagram, actors and users of the system and their relationship with the Garcon system as a whole is pictured. In use case diagram, all possible use cases of the system and their related internal or external actors are pictured. The two most important use cases of the system are explained in detail below the use case diagram.

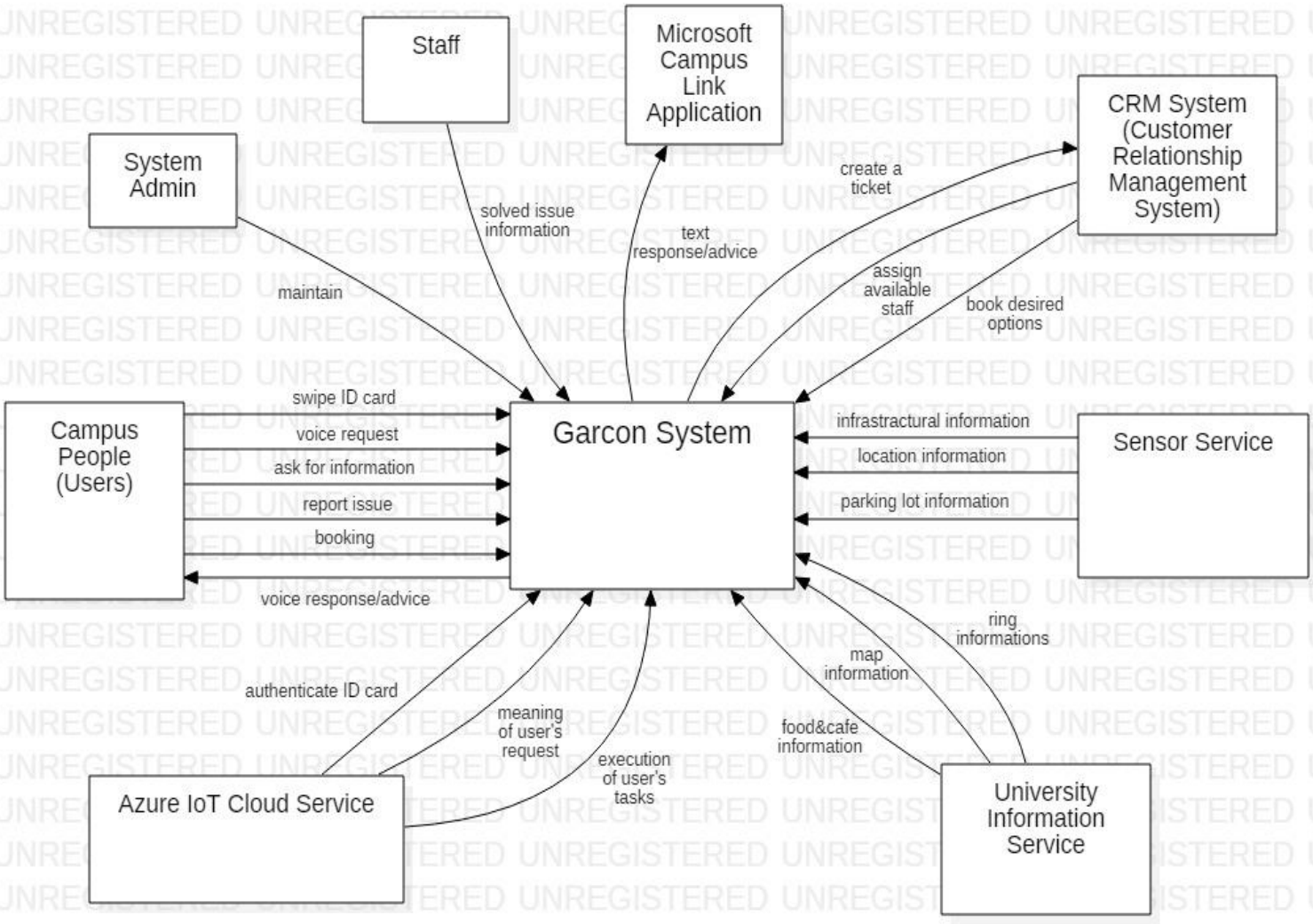

Figure 1: Context Diagram

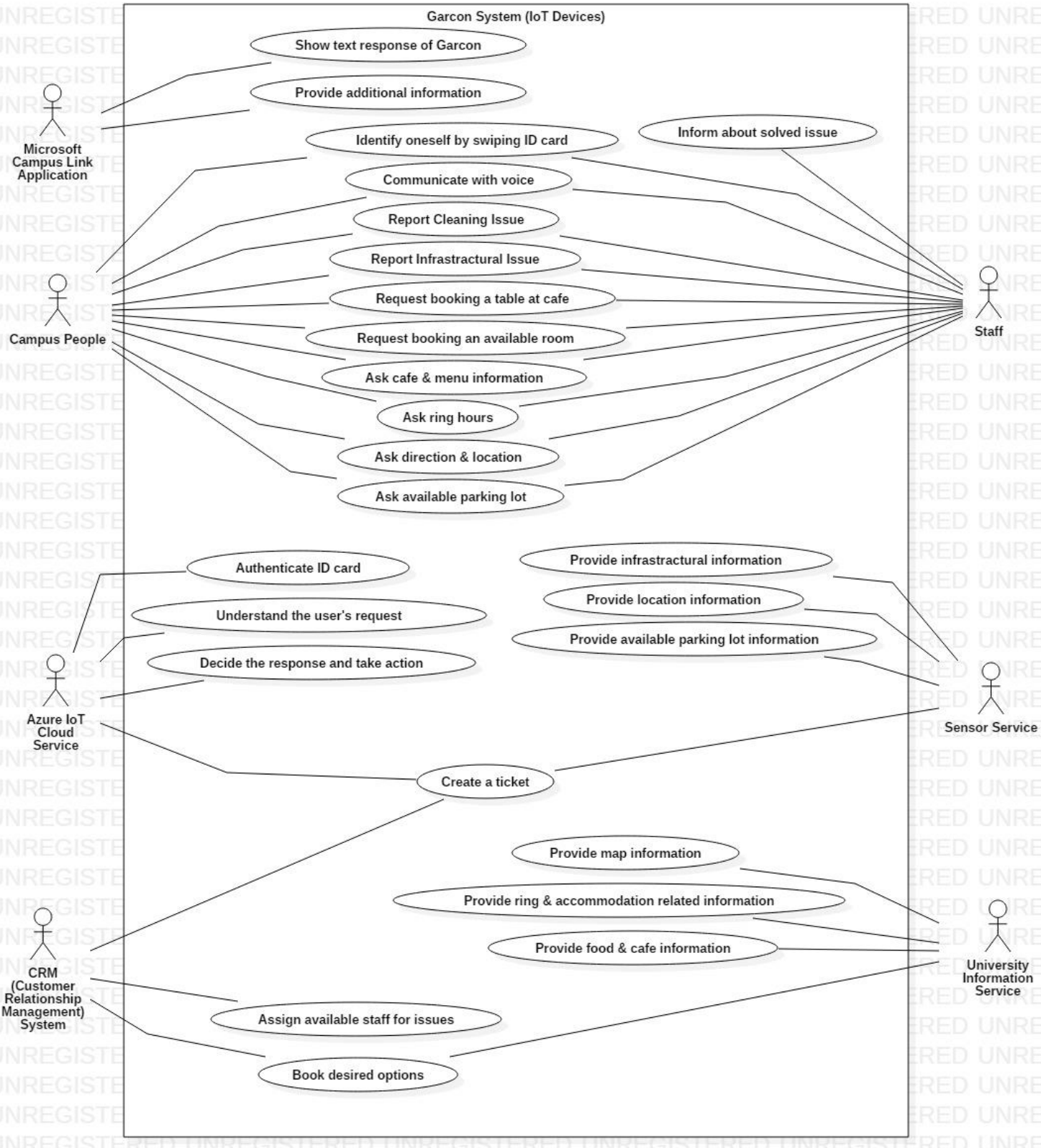

Figure 2: Use Case Diagram

Use case name	Create a ticket
Actors	CRM (Customer Relationship Management System), Azure IoT Cloud Service, Sensor System
Description	If any campus people (students, faculty members, guests with ID card etc.) or any staff reports any kind of issue, Garcon system collects appropriate and necessary informations for these desires and creates a ticket to achieve them.
Data	User voice, location information, infrastractural details of problem (temperature of room, lighting power of building etc.).
Preconditions	Sensor service must be ready to send necessary informations to create a ticket.
Stimulus	Garcon system is notified when voice is translated to text inside the IoT device.
Basic Flow	Step 1 – Garcon system is notified by voice-to-text translator inside the IoT device. Step 2 – Input text is sent to Azure IoT Cloud Service to meaning deduction. Step 3 – If the text is about reporting an issue, Azure IoT Cloud Service collects necessary data (location or technical details of the issue) from Sensor system related to interpreted input text. Step 4 – Azure IoT Cloud Service creates a proper ticket related to issue and triggers an e-mail to CRM system about the ticket. Step 5 – Azure IoT Cloud Service sends information about reponse of the user task (status or result of creating ticket task for this case) and related data to Garcon IoT device to respond user precisely.
Alternative Flow #1	Step 3 – If the text is not about the reporting an issue, another task will be executed by Azure IoT Cloud Service.
Exception Flow	If any sensor hardware malfunctions during gathering data and can not perform its task, it will be recorded as an error.
Postconditions	CRM System can see the ticket and deal with it to solve the issue.

Table 2: Create a ticket function

Use case name	Book desired options
Actors	CRM (Customer Relationship Management System), University Information Service
Description	If any campus people (students, faculty members, guests with ID card etc.) or any staff requests booking for a table at cafe or a room at guesthouse, Garcon system searches available alternatives from the service and if possible books desired places in real-time.
Data	User voice, user ID card number, available places information for booking
Preconditions	User must request a booking by voice from the IoT device.
Stimulus	Garcon IoT device is notified when Cloud service detects a request for booking from the user input.
Basic Flow	Step 1 – IoT device is informed by Cloud service about booking request and desired options of user. Step 2 – Garcon system gather available cafe or guesthouse options from University Information Service according to user's preferences. Step 3 – IoT device informs user with voice about available options. Step 4 – If the user chooses or approves clear option according to Garcon's respond, Garcon system triggers an e-mail to CRM system about user's booking preference. Step 5 – IoT device informs user about the process and directs him/her to Microsoft Campus Link Application to track his/her booking status. Step 6 – CRM System books desired place for the user.
Alternative Flow	–
Exception Flow	Step 4 – If the user does not respond in limited time for approving or choosing clear option according to Garcon's respond, user's ID card authentication is terminated.
Postconditions	User can see his/her booking status from Microsoft Campus Link Application.

Table 3: Book desired options function

4.2 Composition View

This viewpoint shows the components and subcomponents of the system from a top-level perspective. More explained information will be at the component's relevant sections.

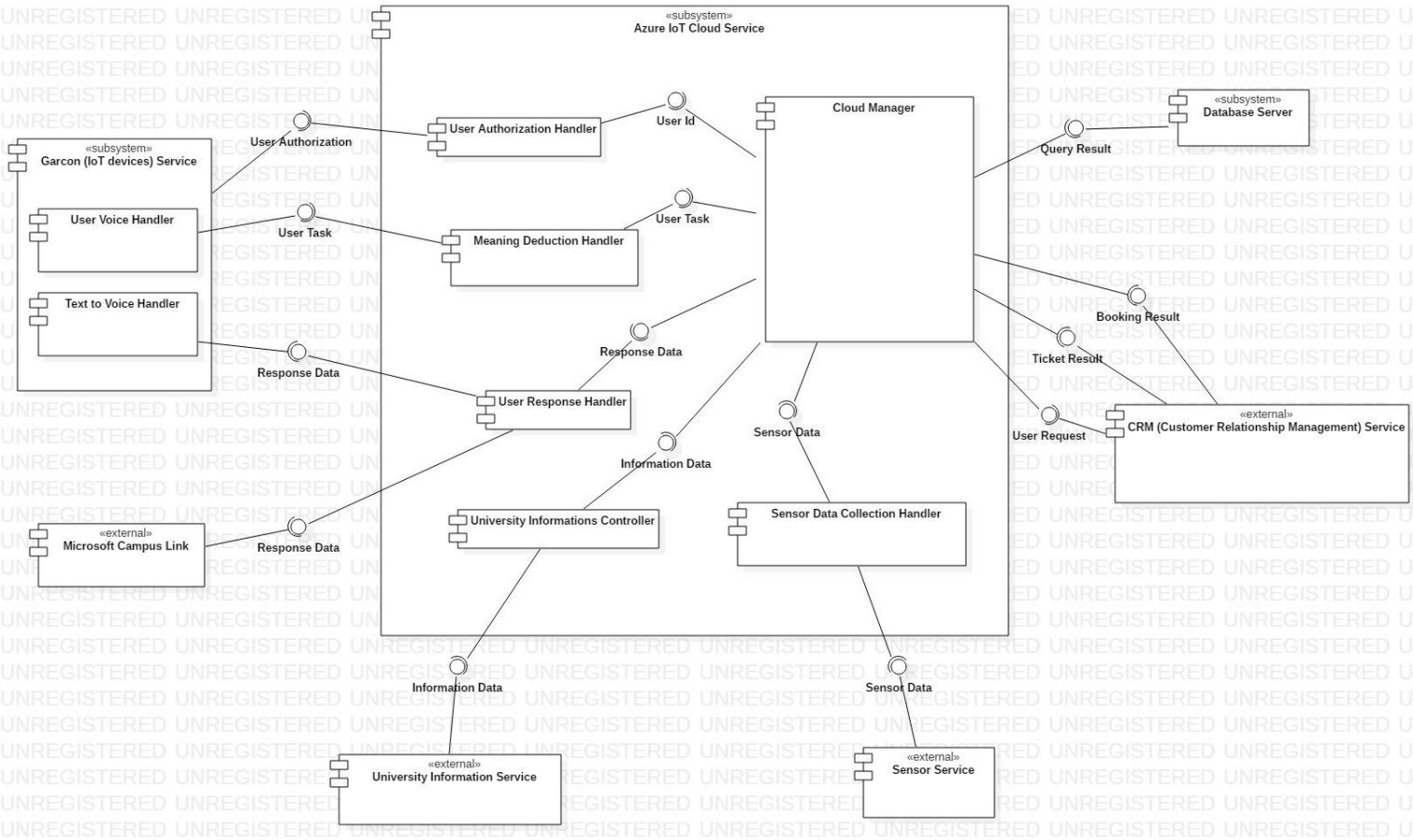

Figure 3: Component Diagram

Design Rationale:

- The most significant component of our system is Azure IoT Cloud Service. Actually, this is the main component and it realizes majority of the jobs for our system. It simply consists of different components, and these smaller components communicate with the remaining external or subsystem components to make system functional as a whole.
- The core subcomponent of Azure IoT is Cloud Manager. It is mainly responsible for providing collaboration with among other Azure IoT subcomponents and realizing database related operations. Other Azure IoT subcomponents' specific jobs are resulted and returned back at this component. It also directly communicates with outer CRM service for ticket and booking related user requests.
- User Authorization Handler component is responsible for realizing authentication and authorization related jobs by using User Id.

- Meaning Deduction Handler is one of the most important component of our system since it decides what user wants from the system by realizing Natural Language Processing basically.
- User Response Handler is responsible for creating and distributing the system response to the user. It directly communicates with Garcon (IoT devices) service and Microsoft Campus Link Service for this purpose.
- University Informations Controller is another component for gathering university related information data from external University Information Service component.
- Sensor Data Collection Handler is responsible for gathering information from Sensor Service which communicates all sensors located at different points of the campus. It also responsible for maintenance and health of sensor system.
- Garcon (IoT devices) Service is another subsystem of our system and it is responsible for realizing in-device operations like voice-to-text translation and maintenance of the devices.
- CRM (Customer Relationship Management) Service is responsible for notifying real people (stuff, system admins etc.) about pending tasks (booking a table, cleaning a classroom ticket etc.) as soon as possible according to commands that are taken from Azure IoT Cloud components. Result or status of these tasks are then reported to Azure Cloud Manager.

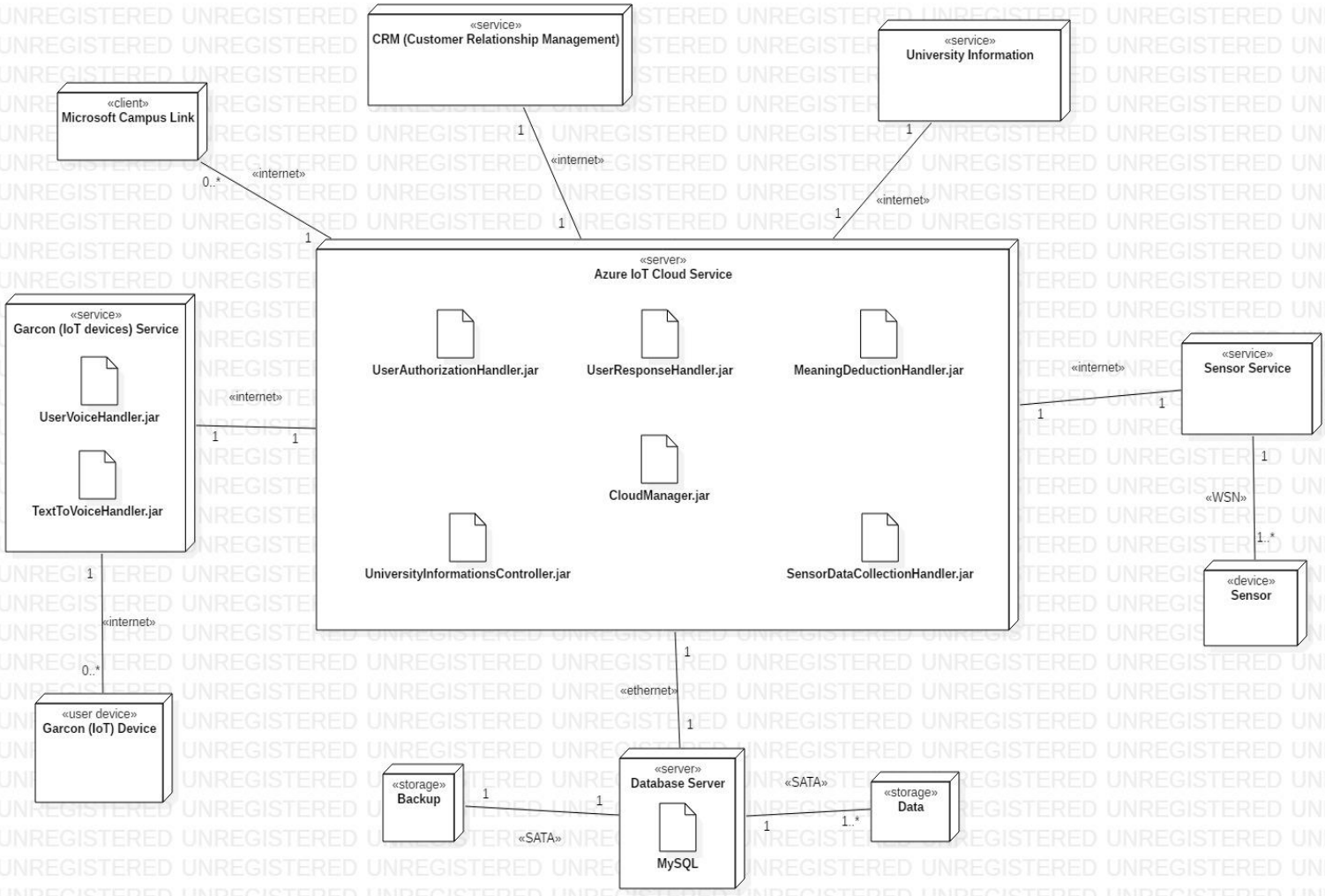

Figure 4: Deployment Diagram

Design Rationale:

- Since our project is basically an IoT project, all external components and other subsystems communicate with our Azure IoT Cloud service by internet.
- Internet based communications are realized with encrypted protocols due to security issues.
- About sensor system, there is a one Sensor Service that can gather data from all sensors from the campus. Although sensor service provide data to Cloud by internet, WSN (Wireless Sensor Network) are used between sensor devices and sensor service as a subset of IoT technology with faster manner.
- There will be two different database storage. One for the backup, and other for our regular data (user, ticket, booking, transaction log etc.).
- MySQL will be used as a relational database management system because of its easy-to-use properties.

4.3 Information View

In this view, system's classes that can affect (create, read, update or delete) any persistent data are explained in detail. Their attributes and methods will be examined according to their effects on persistent data of our system. Moreover, organization and relationship of our data is pictured as Database Class Diagram.

4.3.1 Service Interfaces

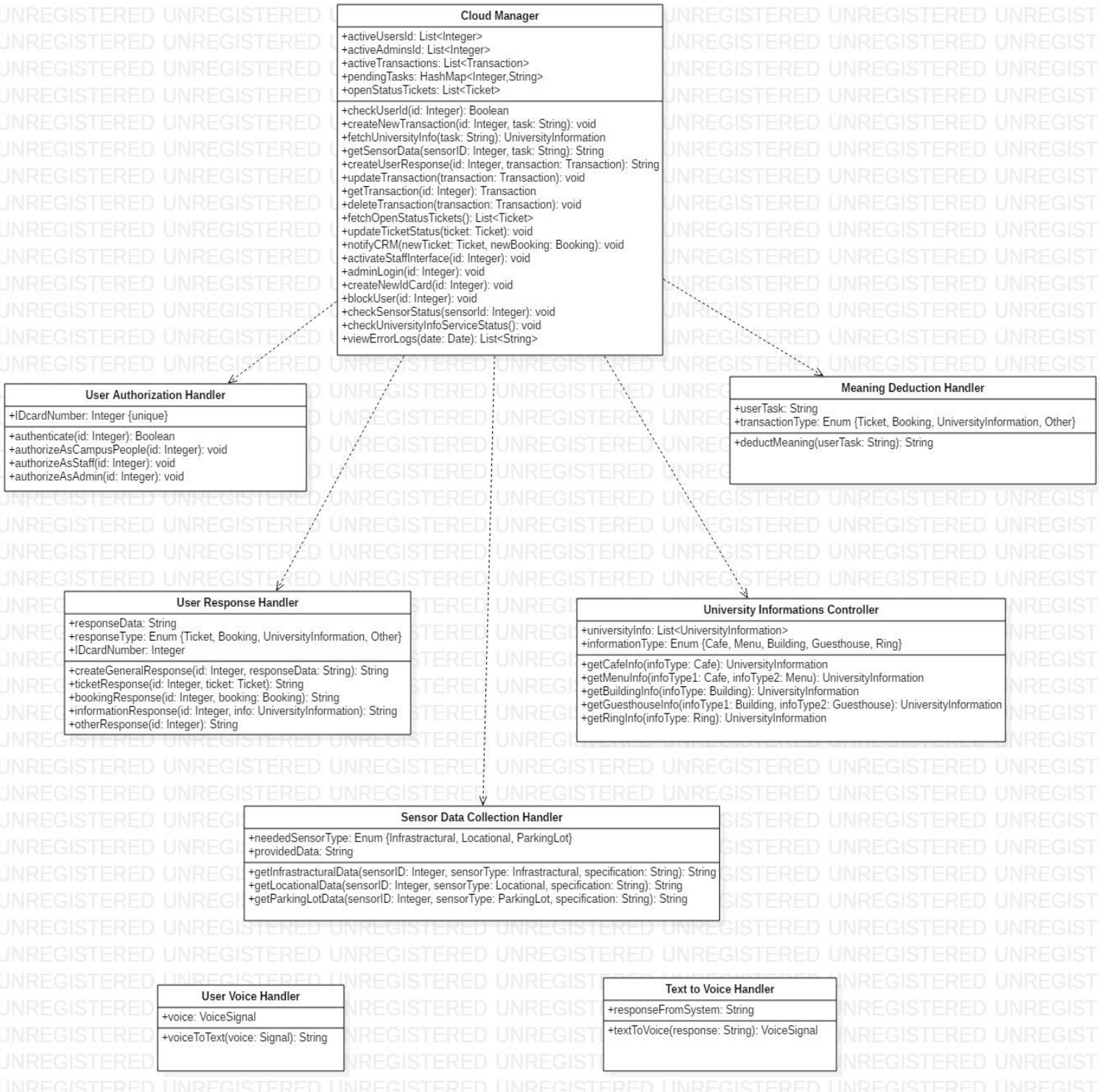

Figure 5: Service Interfaces Class Diagram

Operation	Description
checkUserId	Controls whether the current user's Id is registered or not in the database.
createNewTransaction	Creates new transaction according to user's current task.
fetchUniversityInfo	Triggers our University Informations Controller to get needed data from University Information Service for a specific transaction.
getSensorData	Triggers our Sensor Data Collection Handler to get needed data from Sensor Service for a specific transaction.
createUserResponse	Triggers our User Response Handler and provides a general template for response of the system to the user.
updateTransaction	Updates specified transaction in the database.
getTransaction	Gets specified transaction from the database.
deleteTransaction	Deletes specified transaction from the database.
fetchOpenStatusTickets	Gets open status tickets from the database to show them in the staff interface of Garcon devices.
updateTicketStatus	Updates status of the ticket according to staff's notification for that specific ticket.
notifyCRM	Notifies external CRM (Customer Relationship Management) service if there is a pending booking or creating ticket request.

activateStaffInterface	Activates staff interface at the Garcon IoT devices for specified staff.
adminLogin	Makes ready the Garcon to listen administrative commands and react them as expected if the current user of any device is admin.
createNewIdCard	Creates new user entity in the database. This is one of the administrative commands and can be used only by system admins.
blockUser	Blocks user's ID card from the system at the database. It can be used only by system admins.
checkSensorStatus	Checks status of sensors placed at campus. It can be used only by system admins.
checkUniversityInfoServiceStatus	Checks status of University Information Service. It can be used only by system admins.
viewErrorLogs	Views error logs that can be occur at any transaction. It can be used only by system admins.
authenticate	Authenticates user according to data gathered from Cloud Manager.
authorizeAsCampusPeople	Authorizes current user as an ordinary campus person if its ID is registered as Campus People at the database.
authorizeAsStaff	Authorizes current user as a staff if its ID is registered as Staff at the database.
authorizeAsAdmin	Authorizes current user as an admin if its ID is registered as System Admin at the database.
createGeneralResponse	Creates a general response for a user according to gathered template from Cloud Manager with the result of his/her specific task.

ticketResponse	Creates a detailed and proper response for ticket related user tasks.
bookingResponse	Creates a detailed and proper response for booking related user tasks.
informationResponse	Creates a detailed and proper response for information asking related user tasks.
otherResponse	Creates a proper response for any other type of communication with user (e.g. telling a joke case).
getInfrastracturalData	Gets infrastractural data from specified sensor.
getLocationalData	Gets locational data from specified sensor.
getParkingLotData	Gets parking lot data from specified sensor.
deductMeaning	Deducts meaning from the text of user's request and informs Cloud Manager about new transaction.
getCafeInfo	Gets cafe information from University Information Service.
getMenuInfo	Gets menu information for any cafe from University Information Service.
getBuildingInfo	Gets building information from University Information Service.
getGuesthouseInfo	Gets guesthouse information from University Information Service.

getRingInfo	Gets ring hours information from University Information Service.
voiceToText	Transforms user voice to text inside the Garcon device.
textToVoice	Transforms system text response to voice.

Table 4: Service interfaces operation descriptions

Operation	**Inputs**	**Outputs**	**Exceptions**
checkUserId	1-) User ID	User is registered in the database or not	Database server is not available.
createNewTransaction	1-) User ID 2-) User Task	–	Sensor service or University Information Service or CRM is not available. Database server is not available.
fetchUniversityInfo	1-) User Task	University Information	University Information Service is not available.
getSensorData	1-) Sensor ID 2-) User Task	Sensor's provided data	Sensor service is not available.
createUserResponse	1-) User ID 2-) Transaction	User response	Garcon IoT service is not available.
updateTransaction	1-) Transaction	–	Sensor service or University Information Service or CRM is not available. Database server is not available.

getTransaction	1-) User ID	Transaction	Database server is not available.
deleteTransaction	1-) Transaction	–	Database server is not available.
fetchOpenStatusTickets	–	Ticket list	Database server is not available.
updateTicketStatus	1-) Ticket	–	Database server is not available.
notifyCRM	1-) Ticket 2-) Booking	–	CRM is not available. Database server is not available.
activateStaffInterface	1-) User ID	–	Garcon IoT service is not available.
adminLogin	1-) User ID	–	Garcon IoT service is not available.
createNewIdCard	1-) User ID	–	Database server is not available.
blockUser	1-) User ID	–	Database server is not available.
checkSensorStatus	1-) Sensor ID	–	Sensor service is not available. Database server is not available.
checkUniversityInfoServiceStatus	–	–	University Information Service is not available. Database server is not available.

viewErrorLogs	1-) Date	List of errors	Database server is not available.
authenticate	1-) User ID	User is allowed to use Garcon or not	Garcon IoT service is not available. Database server is not available.
authorizeAsCampusPeople	1-) User ID	–	Garcon IoT service is not available.
authorizeAsStaff	1-) User ID	–	Garcon IoT service is not available.
authorizeAsAdmin	1-) User ID	–	Garcon IoT service is not available.
createGeneralResponse	1-) User ID 2-) Response data	General user response	Garcon IoT service is not available.
ticketResponse	1-) User ID 2-) Ticket	Specific user response	Garcon IoT service is not available.
bookingResponse	1-) User ID 2-) Booking	Specific user response	Garcon IoT service is not available.
informationResponse	1-) User ID 2-) University Information	Specific user response	Garcon IoT service is not available.
otherResponse	1-) User ID	Specific user response	Garcon IoT service is not available.
getInfrastracturalData	1-) Sensor ID 2-) Sensor Type 3-) Data specification	Needed sensor data	Sensor service is not available.
getLocationalData	1-) Sensor ID 2-) Sensor Type 3-) Data specification	Needed sensor data	Sensor service is not available.

getParkingLotData	1-) Sensor ID 2-) Sensor Type 3-) Data specification	Needed sensor data	Sensor service is not available.
deductMeaning	1-) User Task	User task's meaning	–
getCafeInfo	1-) Info Type	University Information	University Information service is not available.
getMenuInfo	1-) Info Type	University Information	University Information service is not available.
getBuildingInfo	1-) Info Type	University Information	University Information service is not available.
getGuesthouseInfo	1-) Info Type	University Information	University Information service is not available.
getRingInfo	1-) Info Type	University Information	University Information service is not available.
voiceToText	1-) Voice	User task	Garcon IoT service is not available.
textToVoice	1-) User response	Voice	Garcon IoT service is not available.

Table 5: Service interfaces operations

Design Rationale:

- The Cloud Manager class is very important for our Azure Cloud Service (the main service of our project). Therefore, it needs to have various methods to work with other services of the system and other handlers in the Cloud conformably.

- Cloud Manager has one general method to control the work of other specific handler classes in the Cloud basically. It also controls database CRUD operations if there is a need as a result of operations of these specific handler classes.
- Administrative operations and staff related operations are also be handled by Cloud Manager.
- Specific handler classes of Azure IoT Cloud (e.g. University Informations Controller, Meaning Deduction Handler etc.) are responsible for creating a communicate and data-flow with other services of system such as University Information Service, Garcon (IoT devices) service or Sensor Service as depicted in Service Interfaces Class Diagram (Figure 5).
- Operations of other services of the system rely on the operations of corresponding handler classes in the Cloud (e.g. operations of Sensor Service relies on the operations of Sensor Data Collection Handler in the Cloud), and these services come to the action if the need for them is detected by Cloud according to newly created user transactions.

4.3.2 Database (CRUD) Operations

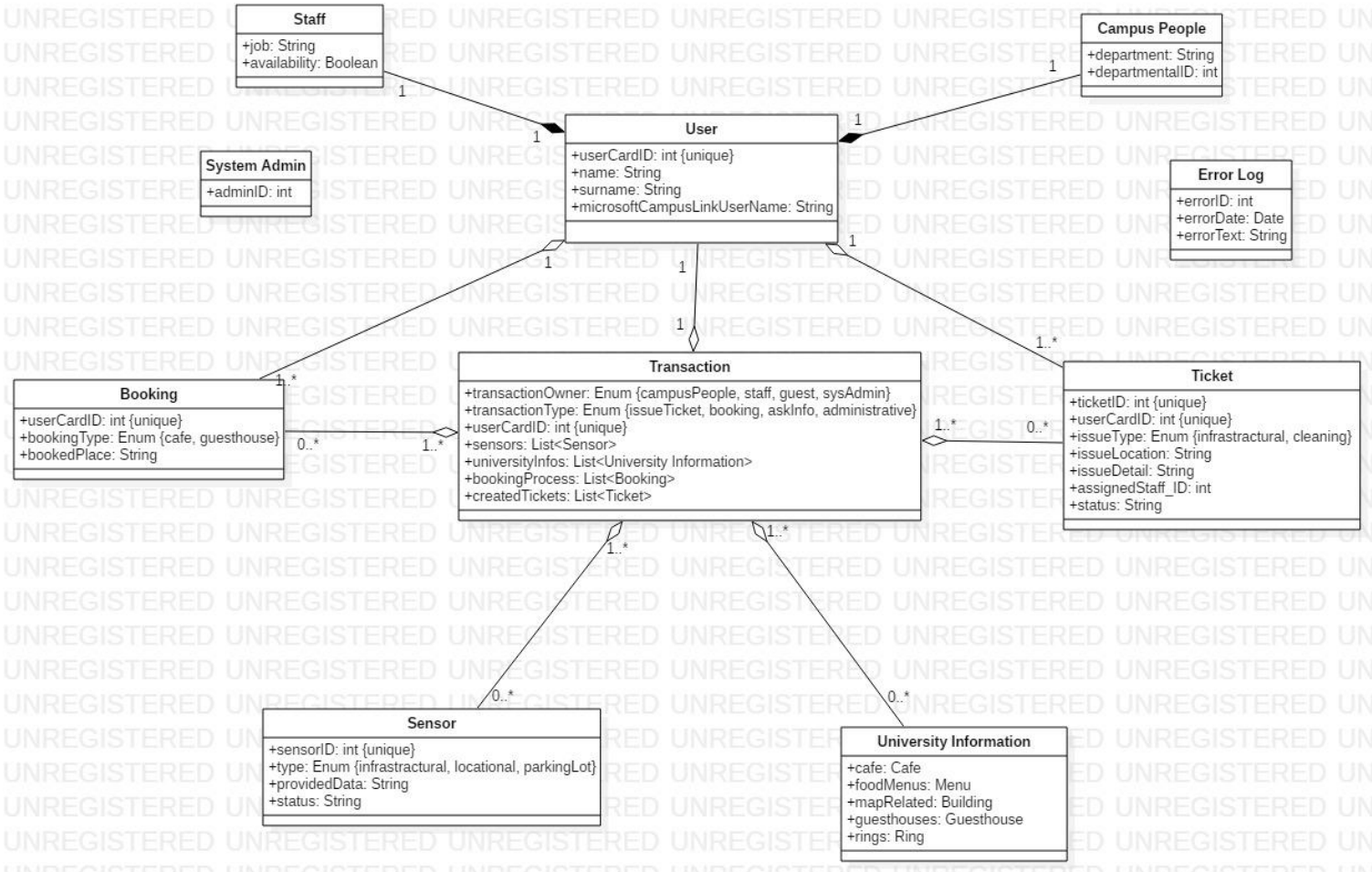

Figure 6: Database Class Diagram

Operation	CRUD Operations
checkUserId	CREATE: READ: User, System Admin UPDATE: DELETE:
createNewTransaction	CREATE: Transaction READ: UPDATE: DELETE:
fetchUniversityInfo	CREATE: University Information READ: University Information UPDATE: DELETE:
getSensorData	CREATE: Sensor READ: Sensor UPDATE: DELETE:
createUserResponse	CREATE: READ: Transaction, User, System Admin, Error Log UPDATE: DELETE:
updateTransaction	CREATE: READ: UPDATE: Transaction DELETE:
getTransaction	CREATE: READ: Transaction UPDATE: DELETE:
deleteTransaction	CREATE: READ: UPDATE: DELETE: Transaction
fetchOpenStatusTickets	CREATE: READ: Ticket UPDATE: Transaction DELETE:
updateTicketStatus	CREATE: READ: UPDATE: Ticket DELETE: Ticket

notifyCRM	CREATE: Ticket, Booking READ: UPDATE: DELETE:
activateStaffInterface	CREATE: READ: Staff, Ticket UPDATE: DELETE:
adminLogin	CREATE: READ: System Admin, Transaction UPDATE: DELETE:
createNewIdCard	CREATE: User READ: UPDATE: DELETE:
blockUser	CREATE: READ: UPDATE: DELETE: User
checkSensorStatus	CREATE: READ: Sensor UPDATE: Sensor DELETE:
viewErrorLogs	CREATE: READ: Error Log UPDATE: DELETE: Error Log

Table 6: CRUD Operations

Design Rationale:

- Since the Cloud Manager is only class that communicates with database directly, all real CRUD operations are done with in Cloud Manager methods according to result or need of other handler methods depicted in Service Interfaces Class Diagram (Figure 5).
- There is no need to keep user response of the system separately for this project. Instead of that, transaction table keeps all transactions with needed details for further investigations that can be done by system admins or staff interface.

- User table keeps all possible end users of the system. Staff and Campus People are weak entities and they have one-to-one relationship with User because they are special type of the user and they are meaningless without the existence of User.
- There will not be any table related to Guest users. Guest's informations shall be kept at User table. However, for system admins, there will be a distinct table that just keeps the admin ID.
- Transaction table keeps every transaction between any end user and the system. Its columns are created according to possible use cases of the system. All columns may not be completely filled for all use cases.
- Booking table keeps user's confirmed and approved booking requests related data and Ticket table keeps data related to issue tickets. Both Booking and Ticket tables has many-to-many relationship with Transaction table since there can be more than one booking process or there can be more than one reporting or viewing ticket process for any transaction (e.g. staff can see all open status ticket list in their transactions).
- Sensor table keeps sensor hardware related data and it has many-to-many relationship with Transaction table.
- University Information table keeps various kind of data such as cafes, menus, buildings or guesthouses. Actually, these are gathered data from University Information Service with related methods, and when there is a frequent need to use these data, University Information table can be used for this purpose.
- Only system admins are able to access all of these tables.
- As a special and crucial point, CRM service is also able to access Staff table and have a full control over Booking table.

4.4 Interface View

In this view, the internal interfaces between the components of the system and the external interfaces between the Garcon's cloud based system and the other services or end users will be specified in detail.

4.4.1 Internal Interfaces

The Interface Between the Database Server and the Cloud Manager: If there is a need for CRUD about data stored in the database, Cloud Manager queries the database. Then, this query is executed in MySQL, and the result is returned back to the Cloud Manager with this interface.

Design Rationale: For majority of transactions between user and system, there will be need to access and handle the data in the database. Thus, this interface is necessary to make our system's other functionalities useful and meaningful at the end.

The Interface Between the User Authorization Handler and the Cloud Manager: Authentication and authorization related processes of the system are handled by User Authorization Handler according to any end user activity comes from Garcon (IoT devices) service. This interface provides necessary ID information flow of any user to be able to check the existence of user ID in database by help of Cloud Manager at the end.

Design Rationale: Since there is no clear connection between User Authorization Handler and Database server, this interface is needed. Moreover, with this interface Cloud Manager will know which users are authenticated at the time being.

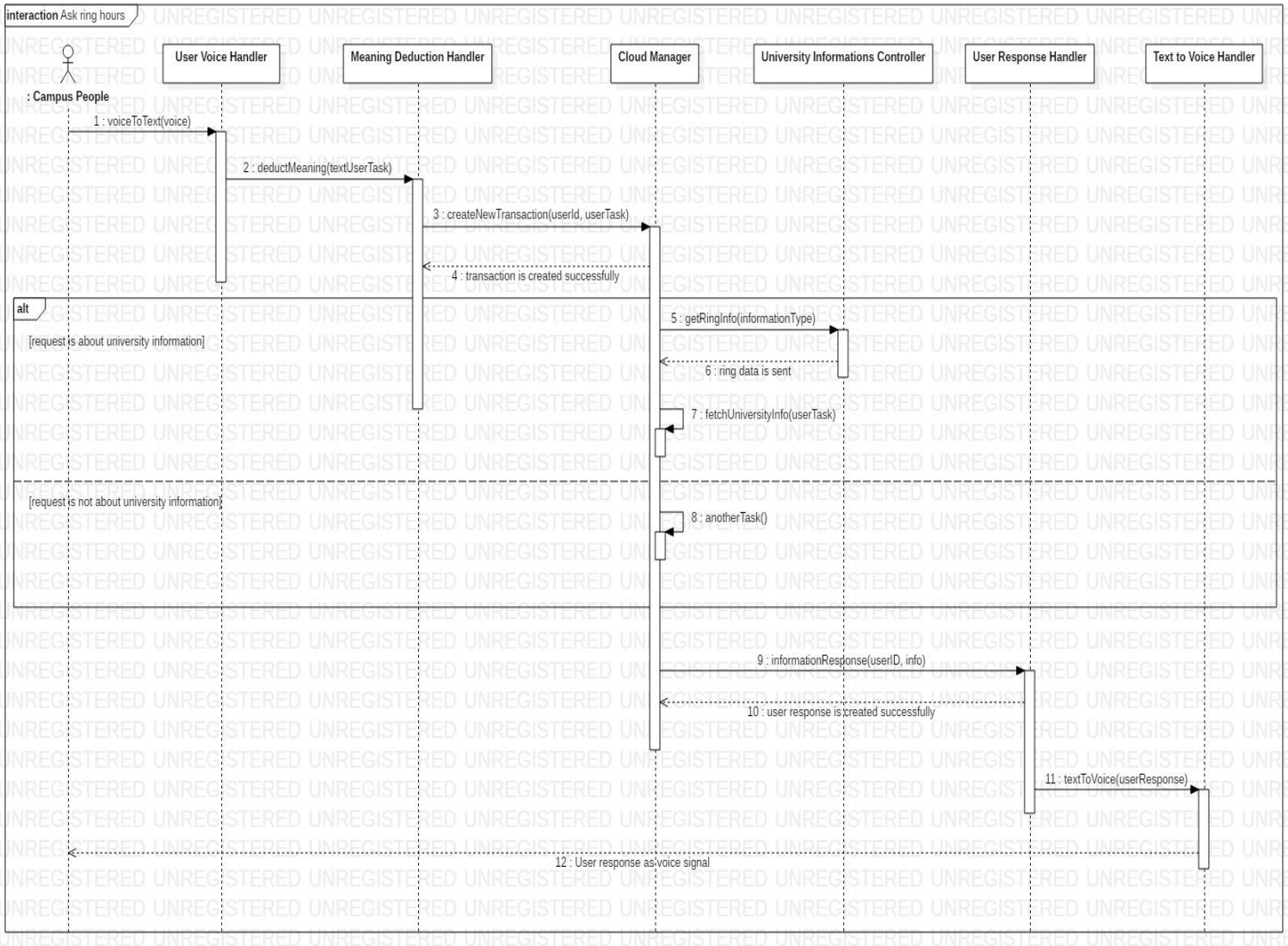

Figure 7: Sequence diagram for ask ring hours use case to show the internal interface between Meaning Deduction Handler and Cloud Manager

The Interface Between the Meaning Deduction Handler and the Cloud Manager: Meaning Deduction Handler transforms user tasks that are gathered from Garcon (IoT devices) service to some clear and distinct forms for the usage of Cloud Manager. Cloud Manager can understand and respond only these special forms (these forms can be Ticket, Booking, University Information or Other. Details can be seen on Service Interfaces Class Diagram, Figure 5) to handle user tasks.

Design Rationale: This interface is one of the most important part for our whole system because user's requests need to be analyzed, understood and then, the results are notified to Cloud Manager to make something useful for the end user.

The Interface Between the User Voice Handler and the Meaning Deduction Handler: User request is transformed from voice to text as it is by the help of User Voice Handler. And then, this user request is needed by Meaning Deduction Handler to analyze and understand the user's intent. This interface provides this kind of communication between these two components of the system.

Design Rationale: This interface is necessary since user request is needed to be carried to Cloud Manager with in special form. And, user request is shaped as a special form through this interface.

The Interface Between the Text to Voice Handler and the User Response Handler: The system's response are created initially at Cloud Manager. Then, User Response Handler is responsible for shaping the initial response to make it more user-friendly and effective. Through this interface, system's response as a text string is conveyed to Text to Voice Handler located in Garcon (IoT devices) service to realize the last step for responding the end user.

Design Rationale: This interface is needed for responding the end user with system's specifically created response to the user.

The Interface Between the User Response Handler and the Cloud Manager: When the system's response to the user is ready, this interface will be used. Cloud Manager initially creates a template for the response as a result of its various response logic for different use case scenarios. And then, this template is gathered through this interface, and the final result is created to convey to the end user by the help of User Response Handler.

Design Rationale: There is a need for this interface because system's response must be examined and shaped as user-friendly before it is sent to the end user.

The Interface Between the University Informations Controller and the Cloud Manager: This interface will be used when there is a user request related to asking university information. Cloud Manager needs to data from University Information Service for this kind of cases. Through this interface, Cloud Manager requests needed data with the help of University Informations Controller.

Design Rationale: This interface is needed because there is a need for specification among University Information Service data. Which kind of data is needed, and how they are fetched are controlled by University Informations Controller. However, the final result must be gathered at Cloud Manager. Thus, this interface will be needed.

The Interface Between the Sensor Data Collection Handler and the Cloud Manager: This interface will be used while creating user response by Cloud Manager. For some user requests, there will be need for gathering data from sensors located in the campus. Cloud Manager needs to this interface to trigger Sensor Service. All communication processes with Sensor Service are done by the help of Sensor Data Collection Handler according to given commands from Cloud Manager through this interface.

Design Rationale: This interface is needed because Cloud Manager needs to specific data for different use case scenarios, and this specification can be done with Sensor Data Collection Handler methods.

4.4.2 External Interfaces

4.4.2.1 User Interfaces

General Garcon (Campus People) Interface: Garcon interface offers just a simple informative screen for all campus people as seen in the Figure 8. It shows a message at the top of the screen to inform users about what he/she can do with this interesting device (e.g. “Issues with the building? Let us know!” message in Figure 8). Also, a picture is shown to tell the user, he/she must first swipe his/her ID card and identify himself/herself before starting to communicate with the device. In fact, this design is quite sufficient when thinking about what Garcon can do for its users because its real power comes from its ability to be activated by sound.

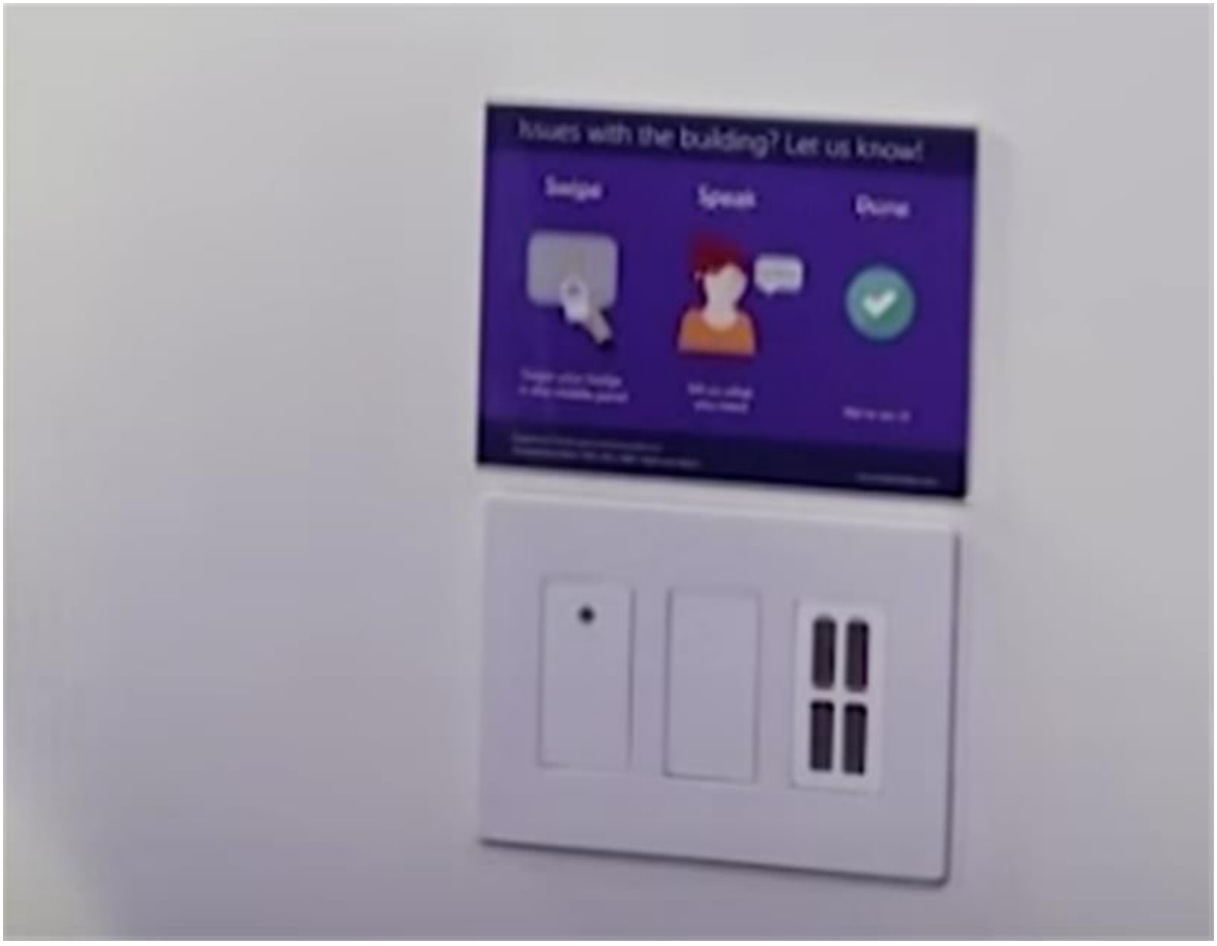

Figure 8: Garcon IoT device main screen

Microsoft Campus Link Interface: This interface is one of the main assistant of Garcon devices because Garcon users can reach extra informations and advices in addition to Garcon's responses through Microsoft Campus Link interface. Actually, when a user asks something to these devices, Garcon generally responds him/her and directs him/her to Microsoft Campus Link Interface to gain more information about the topic. Garcon also notifies this application about user's all requests from the Garcon. Users can reach Microsoft Campus Link Application from mobile devices and when they join with their usernames and passwords, they can automatically see their last interactions with Garcon and also, they can see Garcon's response and extra additional information related to this response (Figure 9). Furthermore, from the perspective of reporting issue and booking capabilities of Garcon, users can trace their reports' and booking requests' status from this application.

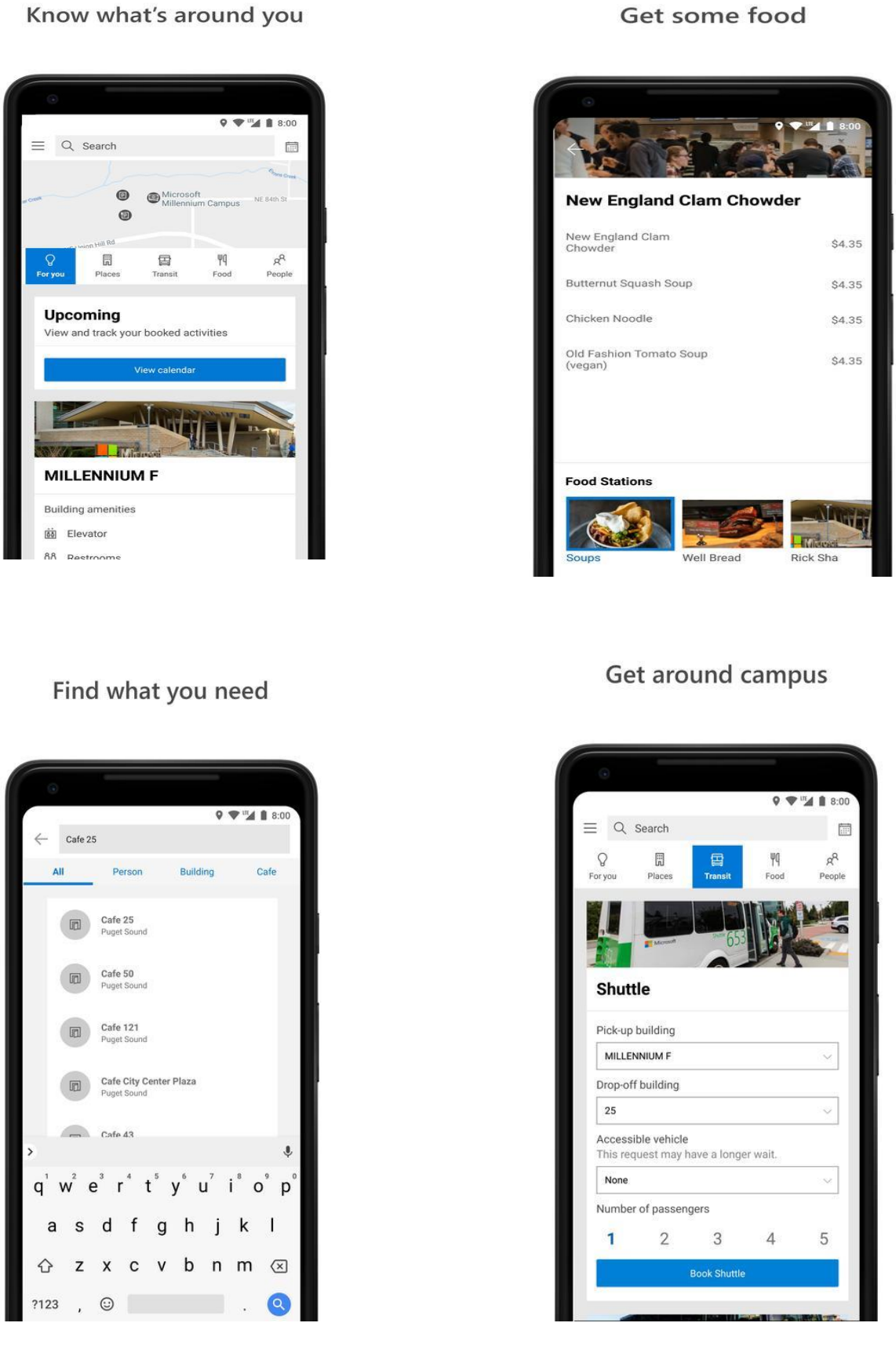

Figure 9: Microsoft Campus Link Application Interface

Staff Interface: Garcon devices also have another special interface which is “Staff Interface”. This interface can be active only when any staff swipes his/her ID card to the device, and then at the first stage, these users are welcomed with the screen that shows the open status issue tickets and the device asks whether the user (who is also staff in fact) have solved any one of these issues or not. And then, of course, staff can inform the device about solving any issue or not through this interface. After this stage, staff can also use Garcon with General Garcon interface.

4.4.2.2 System Interfaces

The Interface Between the User Response Handler and Microsoft Campus Link Application: When a final user response is created from User Response Handler, Microsoft Campus Link account of the user also needs to this response. Through this interface, User Response Handler sends response data to Microsoft Campus Link application.

Design Rationale: Since user needs to trace his/her request’s results or status (this is needed especially for booking or ticket related requests), user should see Garcon’s response and additional information about this response through his/her Microsoft Campus Link application. Hence, there is a need for this interface between User Response Handler of Azure IoT Cloud Service and Microsoft Campus Link application.

The Interface Between the University Informations Controller and University Information Service: University Informations Controller is a bridge between Cloud Manager of Azure IoT Cloud Service and external University Information Service. With this interface, data flow from University Information Service occurs when user requests any university related information.

Design Rationale: Since Azure IoT Cloud Service may need various kind of university related information according to user request, these data must be fetched through this interface. And, since major data handler methods related to university informations are placed in University Informations Controller, this interface must be placed between University Informations Controller of Azure IoT Cloud Service and external University Information Service.

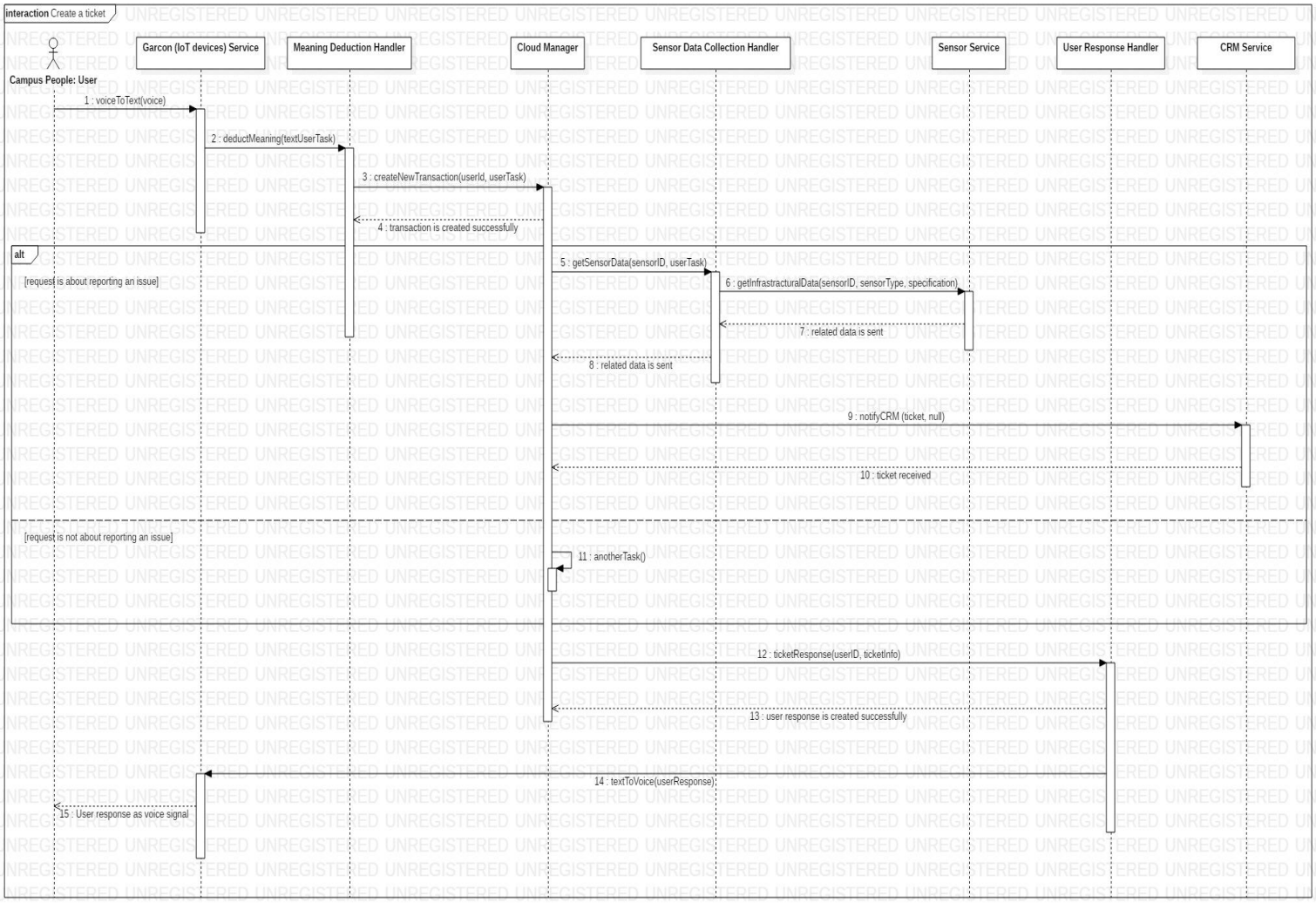

Figure 10: Sequence diagram for create a ticket use case to show the external interface between Sensor Data Collection Handler and Sensor Service

The Interface Between the Sensor Data Collection Handler and the Sensor Service: This interface is provided due to realizing communication between Azure IoT Cloud Service and external Sensor Service from a general perspective. When there is a user request that needs to sensor data, Cloud Manager of Azure IoT Cloud Service triggers a special method placed in Sensor Data Collection Handler. Then, Sensor Data Collection Handler needs to acquire up-to-date data from Sensor Service. Thus, this interface will be provided.

Design Rationale: Cloud Manager of Azure IoT Cloud Service always needs sensor data when there is a user request about reporting an issue as depicted in above figure. Thus, there should be an interface between special Sensor Data Collection Handler of Azure Cloud and external Sensor Service.

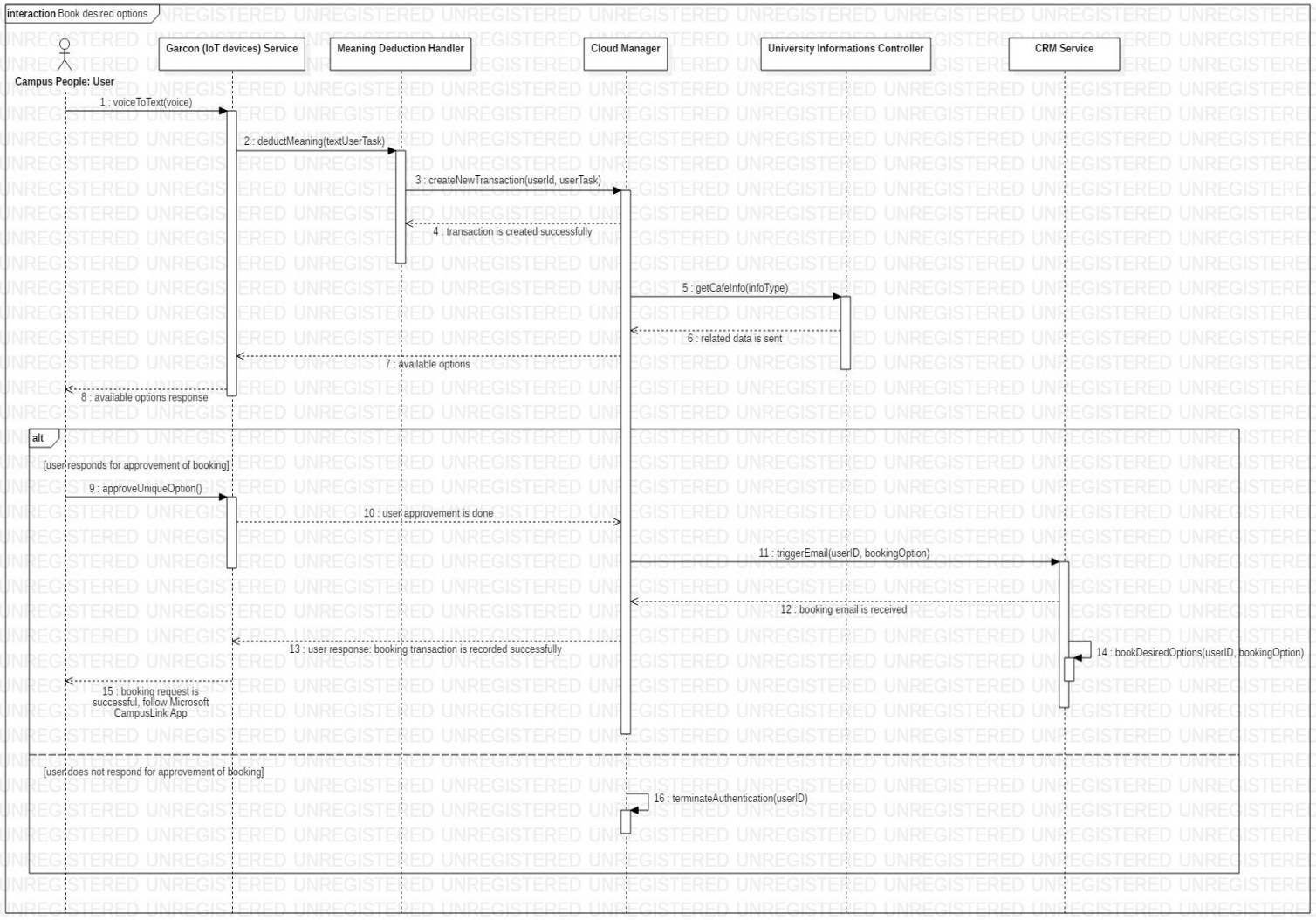

Figure 11: Sequence diagram for book desired options use case to show the external interface between Cloud Manager and CRM Service

The Interface Between the Cloud Manager and the CRM (Customer Relationship Manager) Service: This interface is one of the most important bridge between two important components of whole system. Cloud Manager needs to notify CRM about ticket and booking related user requests. On the other hand, CRM service needs to notify Cloud Manager about status or results of ticket and booking requests according to given data from Cloud. Thus, this interface will be needed.

Design Rationale: The CRM service is a service that allows the user to fulfill his/her wishes in real life. All back-end logic between all components of the system will not make sense without this interface with the CRM service because at the end of the day the system will provide a useful and real result by the help of CRM service.