GARCON

Software Design Description

Ertuğrul Aypek, 2171270
Mustafa Badıllı, 2171296

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Purpose of the System

Microsoft Garcon is an Artificial Intelligence and Internet of Things based smart system. The system is developed by engineers at Microsoft and focuses on making campuses smarter through technology. The project aims to make people's life easier by letting them access useful capabilities both from mobile application and speaker interface on a campus.

1.2 Scope

● System will have a hardware interface that allows users to login with their badges. This interface will get input as voice of the user, translates into text and send the translated text into the cloud in order the information to be processed. The result of the process will be replied to the user as a voice also.

● System will have a mobile application interface that can be accessed by users. Through this interface, users will be allowed to access campus information and their own history and status for processes.

● System will have an interface for campus staff on which they can get tickets, report tickets to related unit of workers and feed-back result of the tickets.

● System will have an interface for system admins that are allowed to authorize and unauthorize all of the users and staff. Both of badge and mobile application can not be used unless approved by system admins.

● System will have an interface for food firms to upload their menus and prices. A food order can be made by users and food firms are allowed to access the current GPS information of the users who ordered a food.

● Users will be able to call a taxi on mobile app and the current GPS information of the caller will be enabled on taxi driver's system interface.

1.3 Stakeholders and their Concerns

Garcon Project is Microsoft's smart campus Project. However, we changed it to be for Metu campus. Thus, stakeholders are users, infrastructure service staff, food company staff, taxi drivers, system admins.

Users : Users are the people who use the facilities of garcon Project. They can create a ticket related to an issue or they may ask garcon for menu from a food company. Moreover, they may call a taxi or see a when a ring will arrive destination in which they waits.

Infrastructure Service Staff : This stakeholder is part of ticket system of garcon. Their relationship with system is that when a ticket is created by user, they see on their system, and they solve the problem and send feed-back to user's mobile app that created ticket.

 Food Company Staff : This stakeholder is part of campus information system of garcon. They get the orders that is given by users and they send menu information to user.

 Taxi drivers : This stakeholder is part of campus information system of garcon. They get the taxi calls done by the user and give response to them using using their system.

 System admins : These are the team members who are responsible of developing and authorization/unauthorizaiton users in Garcon system.

2. References

[1] Alghamdi, A. and Shetty, S. (2016). Survey Toward a Smart Campus Using the Internet of Things. In: *IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*. Vienna: IEEE.

[2] Marti, W. S., Ciptadi, W. P. and Hardyanto, R. H. (2017). Study of Smart Campus Development Using Internet of Things Technology. In: *IOP Conference Series: Materials Science and Engineering.* 190. 012032. 10.1088/1757-899X/190/1/012032.

3. Glossary

Term	Definition
Cloud	A recent technology approach to store huge data such as OneDrive, Google Drive, Dropbox etc.
App	A short form of the word application which is a computer program with some interfaces.
ID	Short for identity which is used to identify a user.
GPU	Short for Graphics Processing Unit
GPS	Short for Global Positioning System

Table 1: Glossary

4. Architectural Views

4.1 Context View

In the context view, context of the system with all actors will be defined in general. In the context diagram, actors and their interaction with the Garcon will be explained in general terms. Use case diagrams and the detailed explanations of some use cases of the system will be specified below the context diagram.
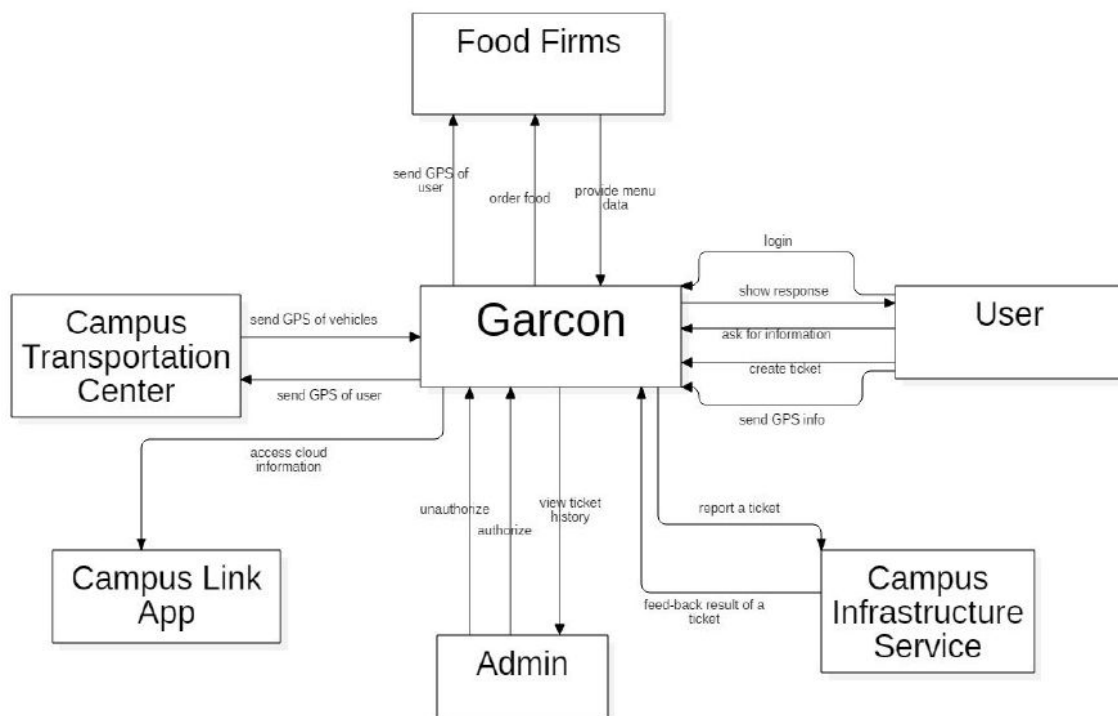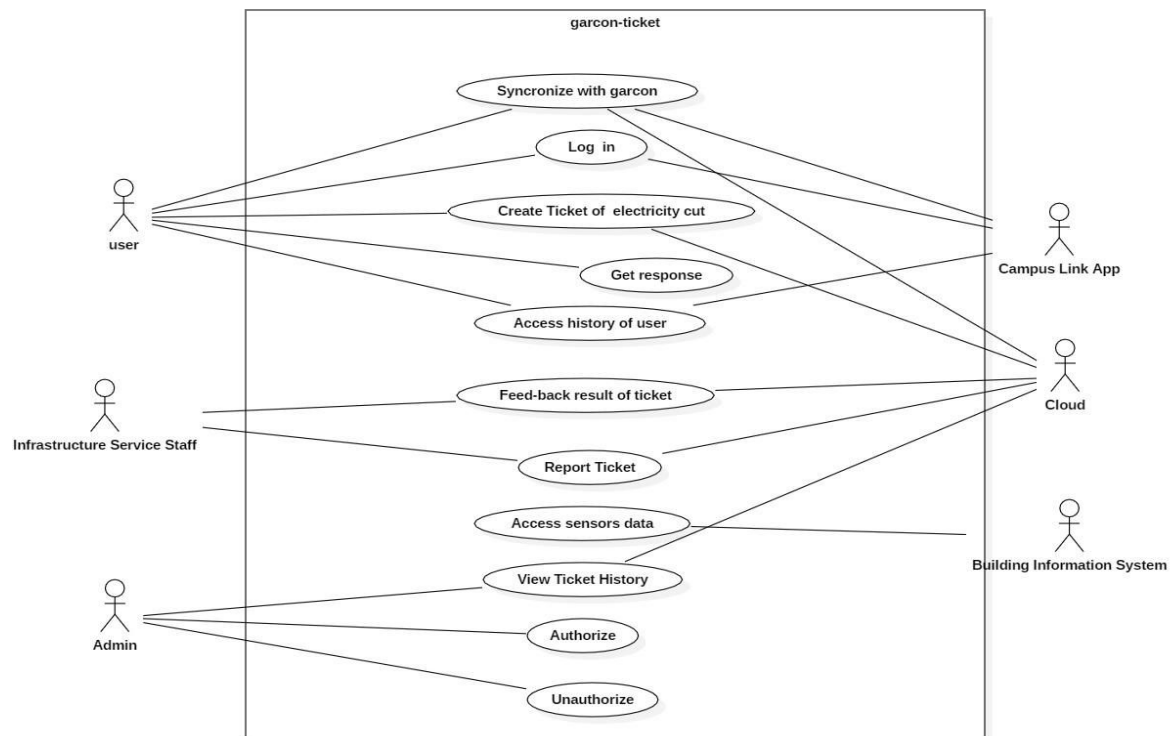

Figure 1: Context Diagram

Figure 2: Use Case Diagram for Ticket System
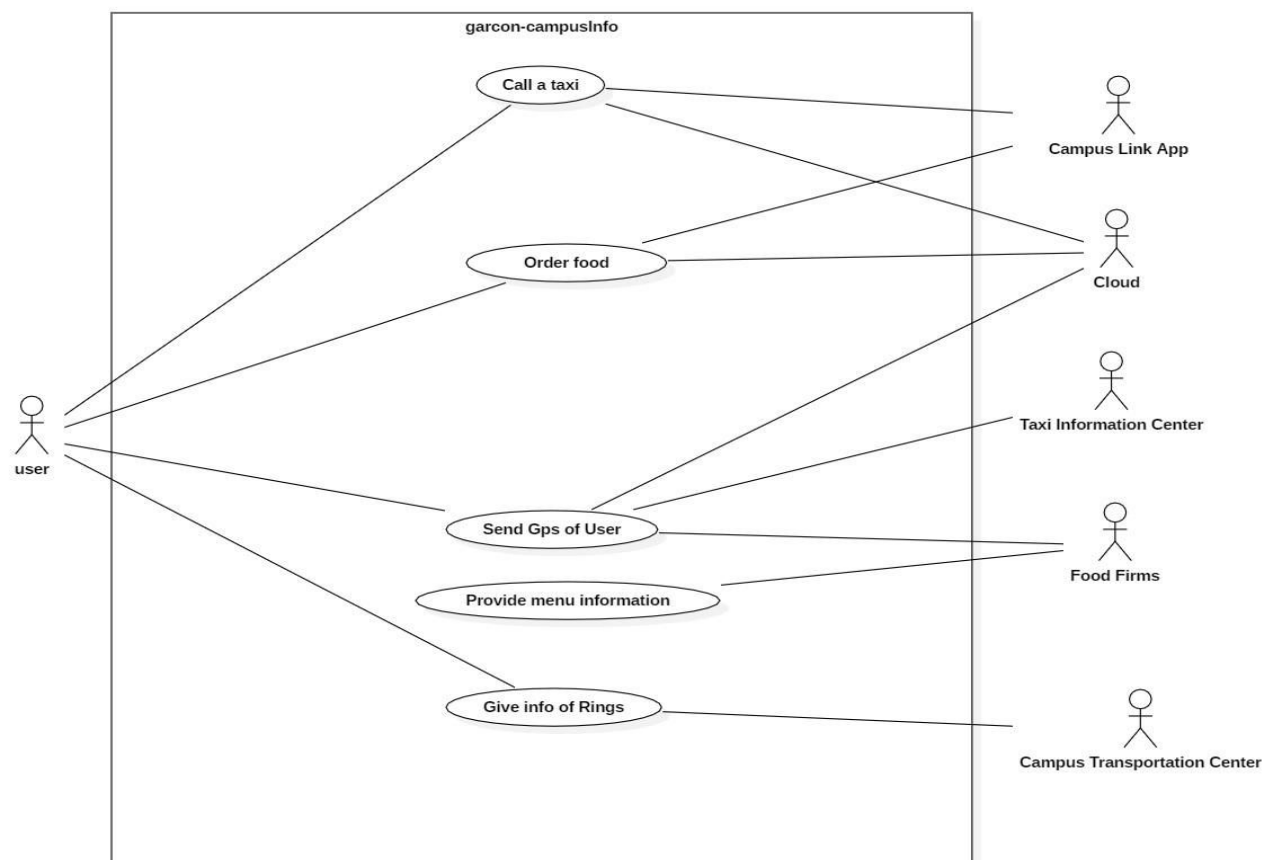
Figure 3: Use case model for campus information system

Use Case Name	Create Ticket of Electricity Cut
Actors	User,Cloud
Description	User informs the system about an issue related to electricity cut. The system get the request, and give feedback such that the problem shall be solved
Data	The information of electricity cut that has been stated by the user
Preconditions	User must be logged in
Stimulus	The system trigger an email to related Infrastructure service staff
Basic Flow	Step 1- User indicates the problem Step 2- The system record it to cloud Step 3- Infrastructure service staff see the problem on the cloud Step 4- The staff solve electricity cut problem
Alternative Flow	Step 2- If a problem occured during sending data to cloud, the process shall be redone until it is done.
Exception Flow	The system indicates the error and ask him for retry
Postconditions	The system returns as "has been recorded"

Table 2: Create Ticket of Electricity Cut function

Use Case Name	Call a Taxi
Actors	User and Campus Link App
Description	Users are allowed to call a taxi by using Campus Link App
Data	Data is the caller's GPS information and if assigned, taxi's GPS information
Preconditions	-
Stimulus	After a call is made by the users, an assignment to a taxi is made and user can see the current location of the taxi.
Basic Flow	Step 1: Call a taxi button is touched by the user Step 2: Current GPS information of the user is sent to Cloud and an assignment of the nearest taxi is made. Step 3: An interface that shows the current location of the taxi on the map and estimated arrival time opens up. Step 4: Interface is closed whenever taxi's and user's locations are very close.
Alternative Flow	Step 3: In case of a failure on taxi assignment due to no taxi available near a threshold distance, user is informed and process is exited.
Exception Flow	In case of no taxi available near a threshold distance, user will be notified.
Postconditions	User may report the driver for some reasons to the system.

Table 3: Call a Taxi Function

Use Case Name	Order food
Actors	User, Cloud and Campus Link App
Description	Users are allowed to order a food from some food firms and to choose to get the food to their current location. Due to security, user's GPS information is sent to food firms via Cloud.
Data	Ordered food and GPS of user (if enabled)
Preconditions	There needs to be some food entries in order users to order.
Stimulus	Using Campus Link app, users will be able to choose a food and order.
Basic Flow	Step 1: User ordered a food Step 2: Ordered food information is sent to food firm's interface Step 3: If enabled, a navigation path appears on food firm's interface Step 4: After the delivery, interface is closed
Alternative Flow	Step 3: If the GPS signal of user is weak, then the user is informed and system asks user to choose a location manually.
Exception Flow	-
Postconditions	In case of unhealthy or uncooked food or any problematic issue, users are able to report to the system.

Table 4: Order Food Function

4.2 Composition View
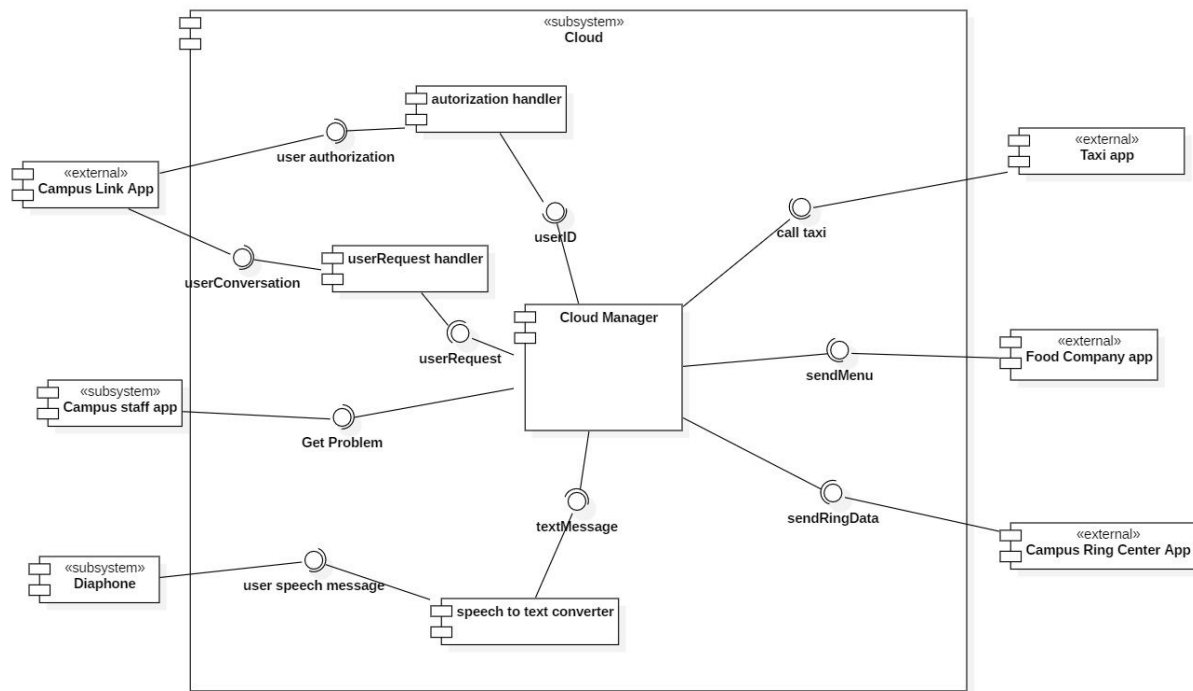
Figure 4 : Component Diagram

Design Rationale:

- Cloud Manager component will be main part of the system as it will be used by the system to communicate with user. It is responsibility is to get request from user or food companies or Campus ring app and send it to either user if it is a request or campus staff if it is related to an issue.

- Authorization handler is responsible for user authorization. That is when a user tries to enter the mobile app it will require a user password, then it will authorize.

- Taxi app is responsible for sending taxi location and if a user calls taxi, it shows that request to taxi driver.

- Food company app is responsible for sending menu to cloud manager such that user will be returned by both diaphone and campus link app immediately.

- Campus ring center app sends data to cloud manager such that when user asked garcon for a ring arrival time, garcon will be able to answer him. Moreover, this operation is possible in campus link app

- Speech to text converter is responsible for converting user's speech to text in order to send data to cloud manager.

- Diaphone is main part of the garcon as user interacts with it directly and it gives response to user immediately. It does this process by sending the user's speech to speech to text converter.

- Campus Staff app is responsible for getting problems that are indicated by user from cloud manager such that related staff will solve the problem.

- Campus Link app is responsible for either keeping records of user's jobs like stating problems, asking menus from food companies or asking ring's arrival time. It allows user to be authorized and it sends user's request to userRequest Handler.

- UserRequest Handler is responsible for sending user's request to cloud manager. This requests either include giving order from a food company or indicating a problem or asking for ring's arrival time.
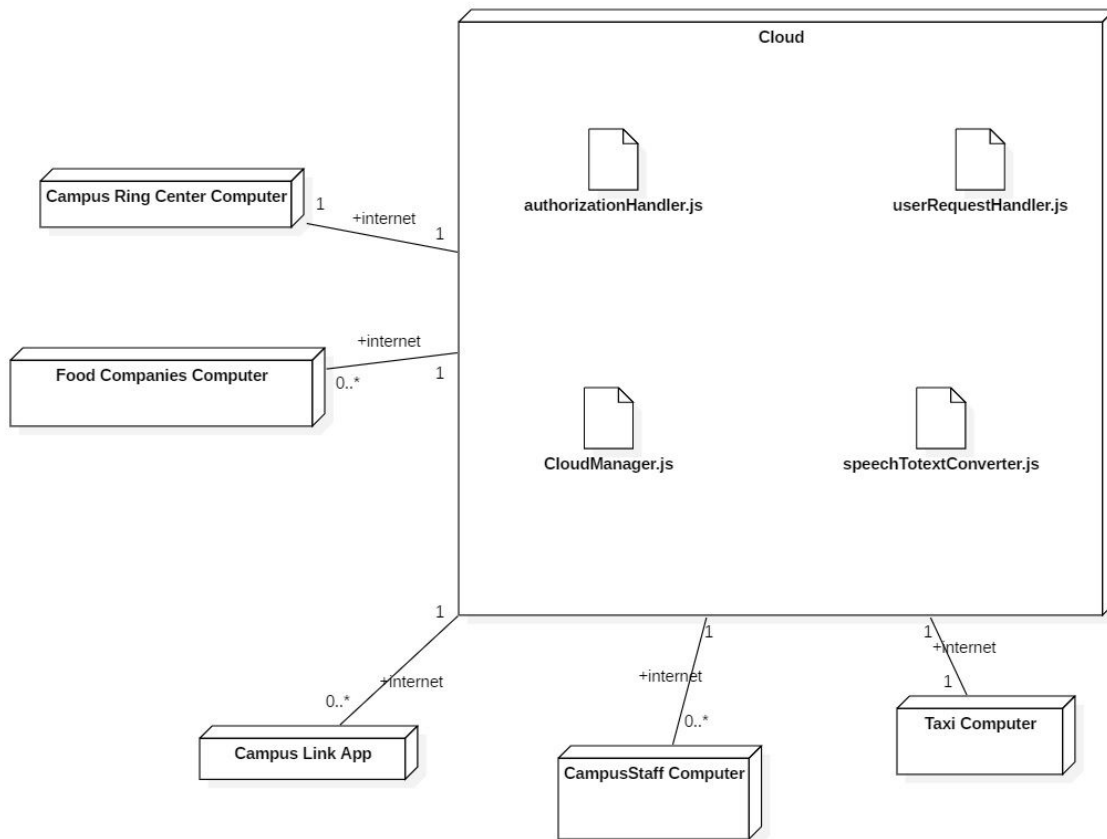
Figure 5: Deployment Diagram

Design Rationale:

- For cloud we used Azure as it is Microsoft Project which will make Garcon easier to adapt it.
- All the files are written in json format to make easier adaptation to new releases of Microsoft upgrades.
- There are alternative cloud servers, to make data transfer safer and to save it from unexpected problems.
- All the internet connections are protected by encrypted protocols like SSL, HTTPS.

4.3 Information View

4.3.1 Database Operations

System Admin

+adminID: int
+adminName: string
+email: string
+password

Ticket Log

+ticketID: int
+assignedStaff: Stafff
+userMessages: List <User Message>
+userID: int

Request Log

+requestID: int
+userAdress: string
+requestType: string
+requestTime: time
+userID: int

User

+userID: int
+userName: string
+email: string
+phoneNo: int
+adress: string

Staff

+staffID: int
+staffName: string
+department: string

Taxi

+driverID: int
+driverName: string
+taxiNo: int
+taxiStation: string

Food company

+companyName: string
+companyAdress: string

Figure 6: Database Class diagram

Operation	CRUD Operations
createTicket	Create: User Read: Staff Update: Delete:
browseTicketHistory	Create: Read: User Update: Delete:
browseTaxis	Create: Read: User Update: Delete:
getRingArrivalTime	Create: Read: User Update: Delete:
getFoodMenu	Create: Food Company Read: User Update: Food Company Delete: Food Company
manageCloud	Create: User, Request log, Ticket log, Food Company, Taxi Read: User, Request log, Ticket log, Food Company, Taxi Update: Delete: Admin

getCampusInfo	Create: Read: User Update: Delete:
signInUserApp	Create: Read: User Update: Delete:
signUpUserApp	Create: User Read: Update: Delete:
sendInfrastructureInfo	Create: Staff Read: User Update: Staff Delete:
convertToText	Create: Read: Request log, Ticket log Update: Delete:
getRequest	Create: Read: Request log, Ticket log Update: Delete:

authorize	Create: Admin
	Read: User
	Update: Admin
	Delete: Admin
unauthorize	Create: Admin
	Read: User
	Update: Admin
	Delete: Admin

Table 5: CRUD operations

Design Rationale:

- Azure will be used for cloud management.
- Every ticket that is created will be kept to see statistical data such that there will be taken precautions in order not to have such problems in future.
- Either user's voice message and its converted version ,text message ,will be kept in cloud for a certain period.
- User's food orders are kept for Food company app such that food company staff will see orders.
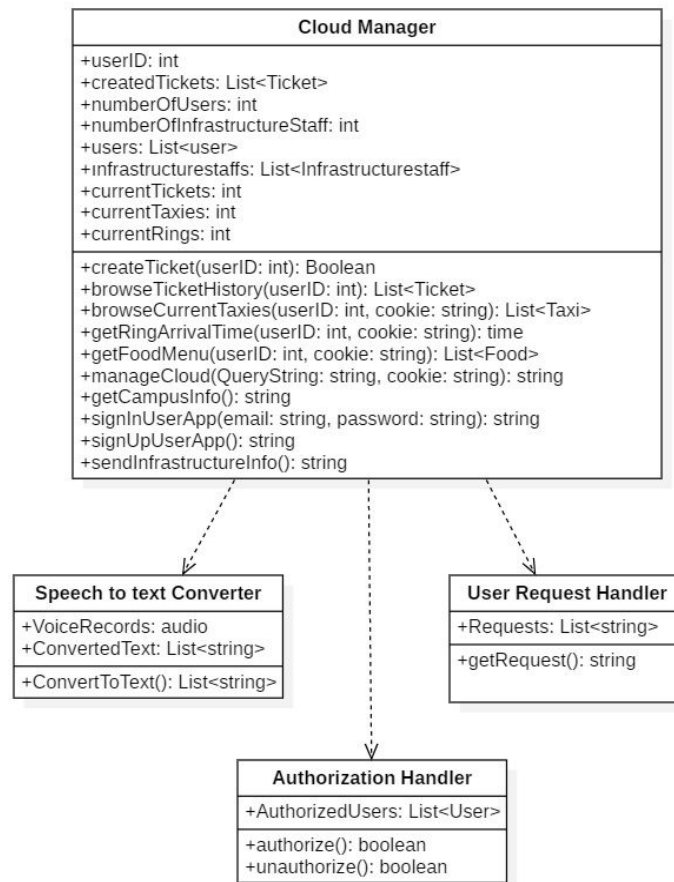
4.4 Interface View

4.4.1 Service Interfaces

Figure 7: interface class diagram

Operation	Description
createTicket	A ticket related to a problem is issued by user.
browseTicketHistory	Ticket history of user can be seen.

browseTaxies	Browses taxis that are nearest to user.
getRingArrivalTime	Ring's arrival time is sent to user.
getFoodMenu	Food menu of nearest or a certain food company's menu is provided.
manageCloud	Manages all operations that are done on cloud.
getCampusInfo	A general information about the campus is given.
signInUserApp	Allows user to sign in user app synchronized with cloud.
signUpUserApp	Allows user to sign up user app synchronized with cloud.
sendInfrastructureInfo	Infrastructure staff sends info related to ticket that user created.
convertToText	User's voice is converted to text message.
getRequest	User request handler gets user's request.
authorize	Allows user to be authorized.
unauthorize	Allows user to be unauthorized.

Table 6: Operation descriptions

Operation	Inputs	Outputs	Exceptions
createTicket	-userID -text	True if successfully created, false otherwise	-Not an authorized user -Connection is lost
browseTicketHistory	-userID	List of tickets	-User has no ticket history -Connection with cloud is lost
browseTaxies	-userID -GPS	List of nearest taxis	-Given GPS is invalid -GPS of taxis are not working
getRingArrivalTime	-userID -GPS	Time to arrive in	- Given GPS is invalid - No arriving ring
getFoodMenu	-userID -food company	List of foods	-No such a food company in campus
manageCloud	-query -adminID	Result of given query	-Cloud connection is lost
getCampusInfo	-userID	A general info as a string	-Cloud connection error happens
signInUserApp	-userID	A message indicating success	-User is not an authorized -Synchronization is lost

signUpUserApp	-userID	A message indicating success	-Synchronization is lost
sendInfrastructureInfo	-userID	Information about created ticket is given	-Cloud connection error happens -Problem has not been solved
convertToText	-userID -user Speech	List of user's sentences	-Cloud connection error occurs -Diaphone error occurs
getRequest	-userID	A message indicating success	-Cloud connection error occurs
authorize	-userID -password	A message indicating success	- Cloud connection error occurs
unauthorize	-userID	A message indicating success	- Cloud connection error occurs

Table 7: Operation design

Design Rationale:

- Cloud manager is main part of the garcon. It is responsible for communication between either user and the infrastructure staff or user and food company or user and taxi driver using cloud interface.
- creatTicket operation will be done using Diaphone asynchronously.
- getRingArrivalTime will be determined by GPS using related ring's velocity.
- All tickets history of user will be stored in database using a list structure.
- browseCurrentTaxies operation will return current taxies that are nearest and sort them in the list in accordance with their possible arrival time.

- getFoodMenu operation will return to the user list of available foods and user may give order to food company during this operation.
- If a ticket has been issued, then infrastructure service staff sends related information using sendInfrastructureInfo operation.
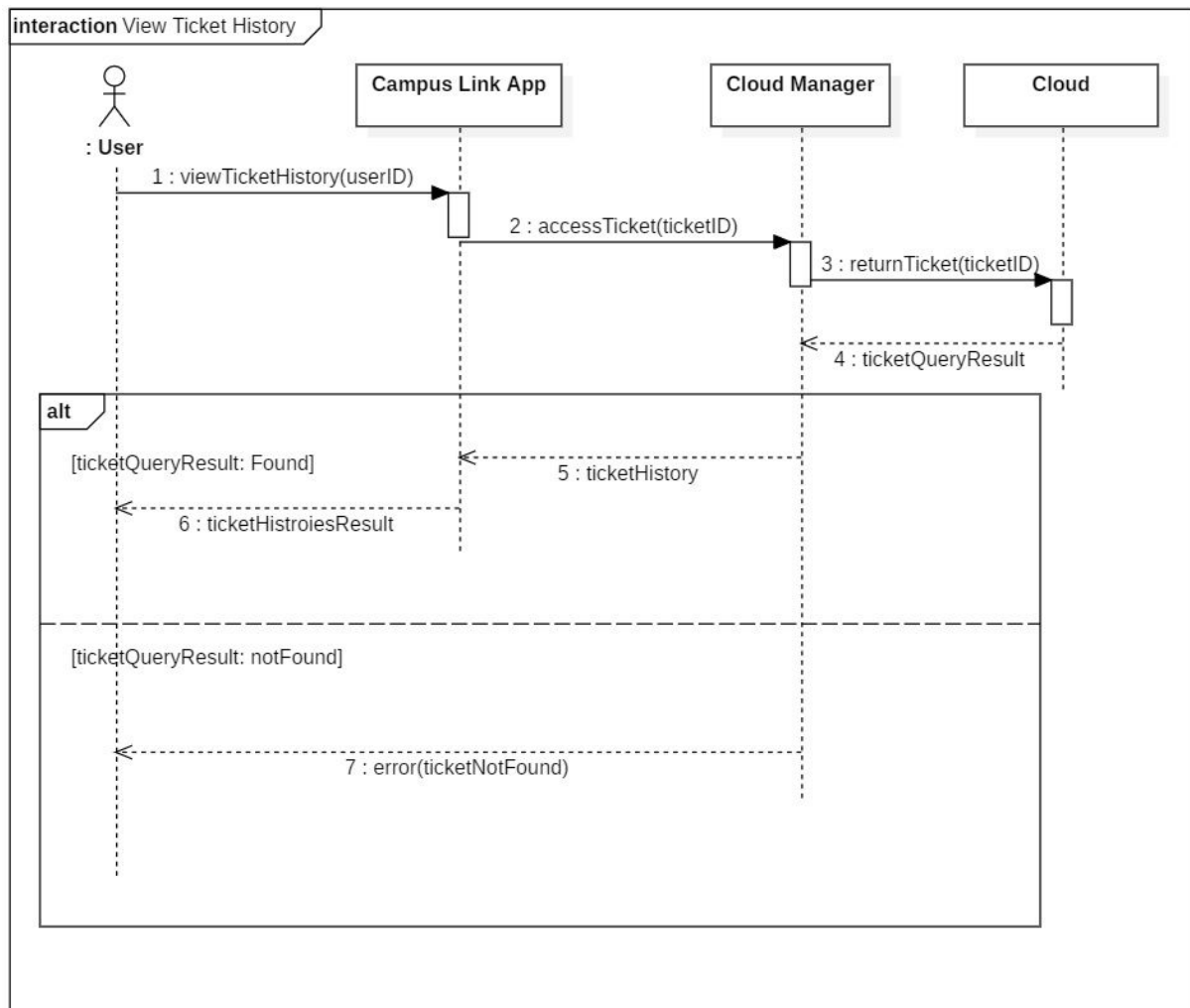

Figure 8: View Ticket History diagram showing the interface between campus link app and cloud manager.

4.4.2 User Interfaces

User Diaphone interface:

      User diaphone interface allows user to interact with garcon. It gets requests from user and there can be a conversation since garcon is an artificial intelligent system.

User diaphone interface is a diaphone which has capability of reading ID cards of user to be able sign in and sending requests. This diaphone interface is common all over the campus and main interaction between user and garcon is done using this interface.

It has ID reader on itself, to make user be able to request. There exists a speaker and a handset to be able speak and listen user.

Design Rationale:

- Similarity with a normal diaphone makes it simple to use.
- Being able to speak and listen provides better functionality for either normal users or disabled users.
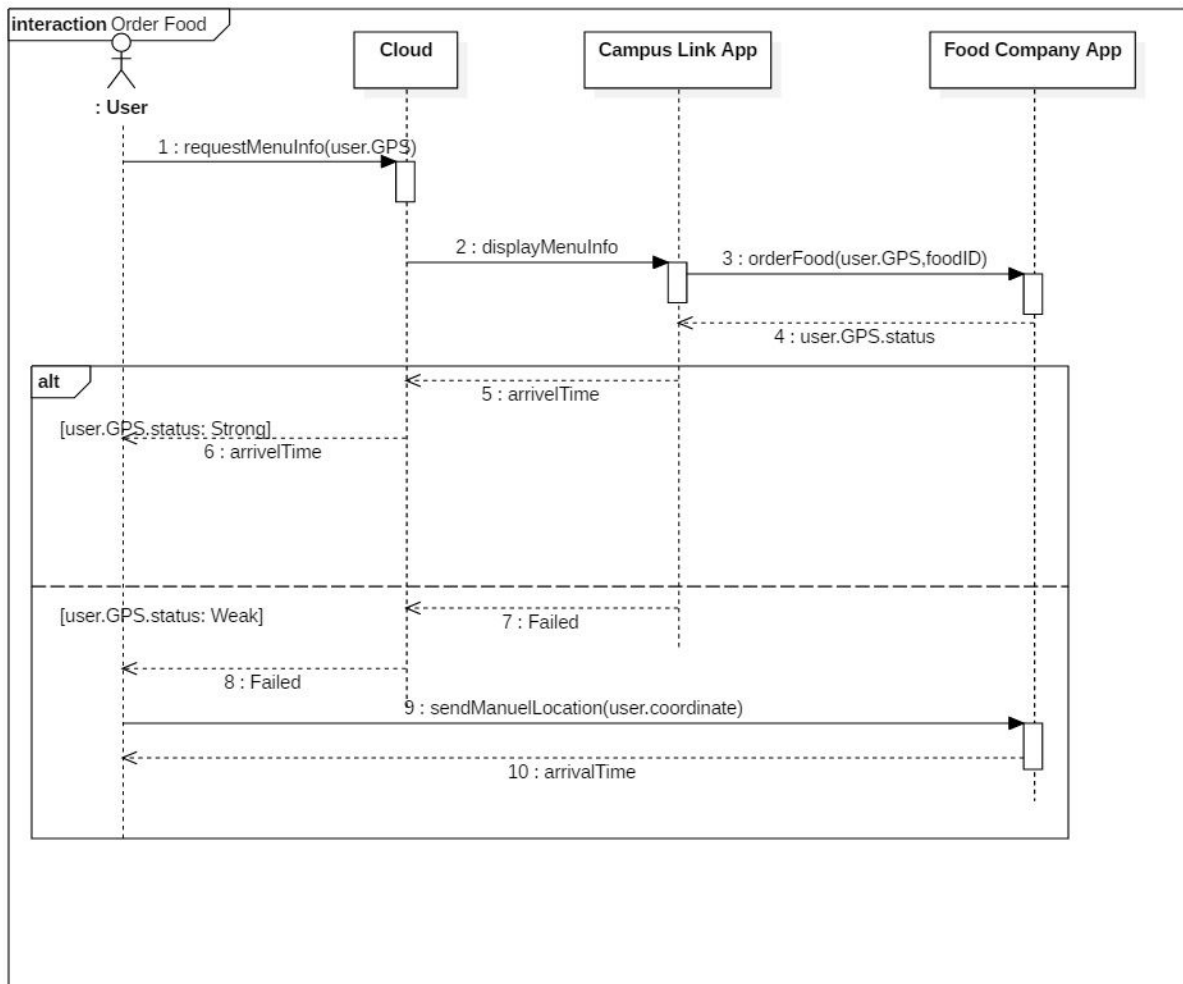- In case of misuse by the user, diaphone will warn the cloud.

Figure 9: Order food diagram showing the interface between campus link app and food company app.

Campus Link App interface:

Campus link app interface allows user to interact with garcon using mobile application. User requests through this app and there can be either food order requests or request for a taxi .

Campus link app interface is an internet-connected app which allows capability of synchronizing user's job with garcon cloud. Moreover, garcon can send the information like food menu through this application. This app is available for all users and main interaction between user and external systems in garcon ,like taxi app, is done using this interface.

Campus Link Application

The students who have newly registered with our university should start student account creation process via mailing to admin@campus.edu

⌂ SIGN IN

Usercode :

Password :

Forgot password ?   SIGN IN

Figure 10: Sign in interface of Campus Link Application

Design Rationale:

- Requiring user id makes it secure.
- Connection through internet allows to be synchronized with other external systems like food company app.

27

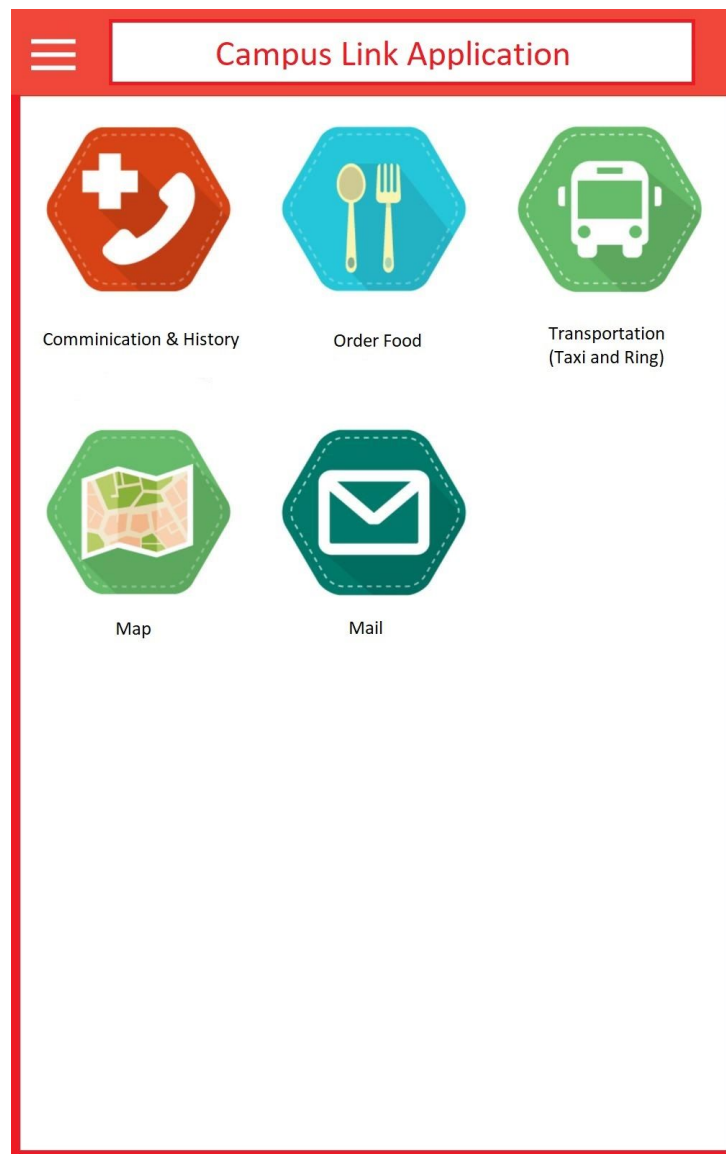- This interface allows user to do all jobs that are possible in garcon.

Figure 11: Menu interface of Campus Link Application

Infrastructure service staff interface:

Infrastructure service staff interface allows the staff to problems all over the campus. This interface composed of two parts: getting ticket, and sending information about related ticket.

Infrastructure service staff sees the campus map and on the campus map, there are main infrastructure service subsystems. In case that a user creates a ticket, there appears red alarm sign on that subsystem. When the staff clicks on it, s/he will see the location of the problem.

There is a part in the interface to send information to user. When user create a ticket, there appears his/her user app mobile address. The staff sends information through the campus link app.

Design Rationale:

- There will be alarm system if a problem is not seen by any staff.
- The interface will be simple thanks to providing campus infrastructure using campus map.
- Usability in terms of user is good since there will be reply after solving the problem.

Taxi driver interface:

Taxi driver interface allows taxi driver to see if a request has been done. This interface will be simple and compatible to taxi. This interface will speak to taxi driver even when s/he drives the taxi. If the taxi driver accepts the request, then s/he gives response to user through this interface.

Design Rationale:

- Taxi driver may see requests either by eyes or by speaker of the interface.
- Speaker opportunity of this interface make this process fast and easier.

Admin interface:

In admin interface, admin will be able to all operations that are done by infrastructure service staff and users' requests or users' tickets. Admin interface may affect other interfaces like food companies app, taxi app interface. Admin interface is able to make changes on the system. If necessary, the garcon system may be stopped at all using this interface.

Design Rationale:

- This interface is able to manipulate cloud.
- Admin interface has relationship with all subsystems and external systems, which allows this interface to control whole garcon system.

4.4.3 Interface Classes

The Interface between the cloud manager and Taxi app:

Cloud manager is responsible for arranging synchronization between taxi app and user. After a user calls taxi, this operation will be directed to taxi app through cloud manager. If the taxi call is successful, user will be informed.

Design Rationale:

- If an error occurs in taxi app, operation is repeated. If the same error still continues, then cloud manager will warn admin.
- If operation is successful, then user will be informed and cloud also will record the operation for user's history.
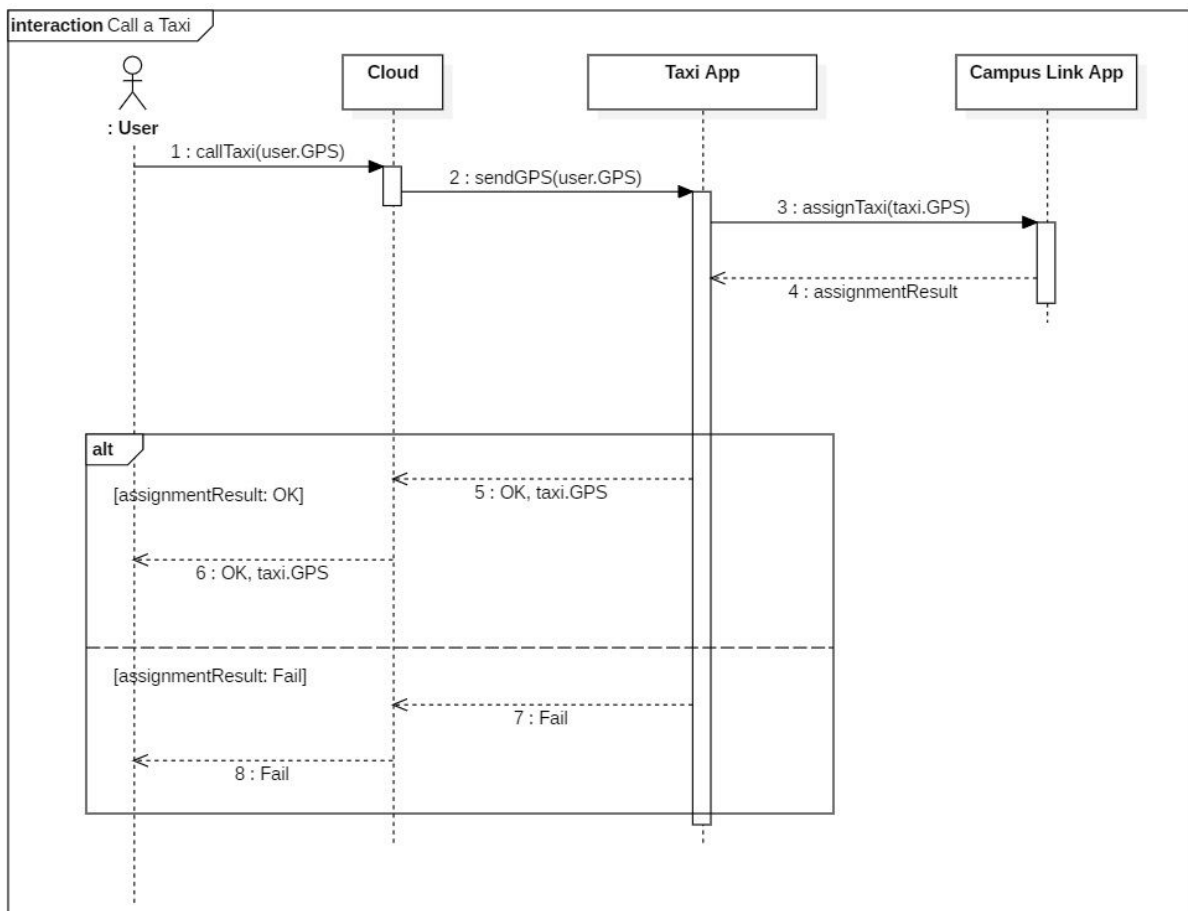

Figure 12: Call a Taxi diagram showing the interface between campus link app and food company app.

The Interface between the cloud manager and Food Company app:

Cloud manager is responsible for synchronization between food company app and user. When a user gives order or requests food menu, this operation will be directed to food company app through cloud manager. If this operation is successful, user will be informed.

Design Rationale:

- The connection between the food company app and cloud manager will be through internet.
- If an error occurs, an error message will be sent to food company app.

The Interface between the cloud manager and Campus Ring Center app:

Campus ring app sends data about the rings to cloud such that cloud will send this information to user. The data that is sent by Campus ring center app may be for a certain ring or a general information about the rings' arrival time.

Design Rationale:

- Campus ring app sends information about the ring arrival time according to the destination that user asks to calculate correct arrival time using ring's velocity.

The Interface between the diaphone and speech to text converter:

Speech to text converter is responsible for converting the user's voice message to text message such that it will be sent to cloud manager. In case of error during operation, the diaphone will be asked again for voice message.

Design Rationale:

- Speech to text converter converts user's voice to text for anything including a normal conversation with Garcon.
- Speech to text converter sends this message to cloud for necessary operations.

- If a problem occurs while converting, error message will be given to diaphone.

The Interface between the cloud manager and user request handler:

User request handler is responsible for arranging requests done by user and send them to cloud manager. Cloud manager decides to connect related system according to request.

Design Rationale:

- User request handler gets user's request by campus link app.
- User request handler arrange coming requests in accordance with related services i.e food order, call taxi.
- In case of error, an error message will be sent to campus link app.

The interface between the cloud manager and speech to text converter:

Speech to text converter is responsible for converting the user's voice message to text message. After converting it sends the data to cloud manager such that cloud manager will do related job.

Design Rationale:

- Cloud manager is responsible for deciding operations according to coming messages from speech to text converter.
- The text messages will be sent in json format.