GARCON

Software Requirements Specification

Ertuğrul Aypek 2171270
Mustafa Badıllı 2171296

Table of Contents

List Of Figures

List of Tables

1. Introduction

1.1. Purpose

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. Product Overview

1.3.1. Product Perspective

Garcon is a huge technology itself which utilizes Internet of Things, Natural Language Processing, Cloud and various sensor technologies. System connects all the people that live on a campus together. Thanks to cloud, Garcon eases people's life by enabling them real-time information about campus using its mobile application. Moreover; for those who have not their mobiles with them, logging in

with badges, the system enables its users to access cloud through a speaker device that is embedded inside wall. This device makes use of Natural Language Processing, converts speech into text and give the output of the system to the user. In terms of Internet of Things, the system has an access to Building Information Centers for each building where the current status of each sensor (fire, water, dangerous gas etc.) is stored.

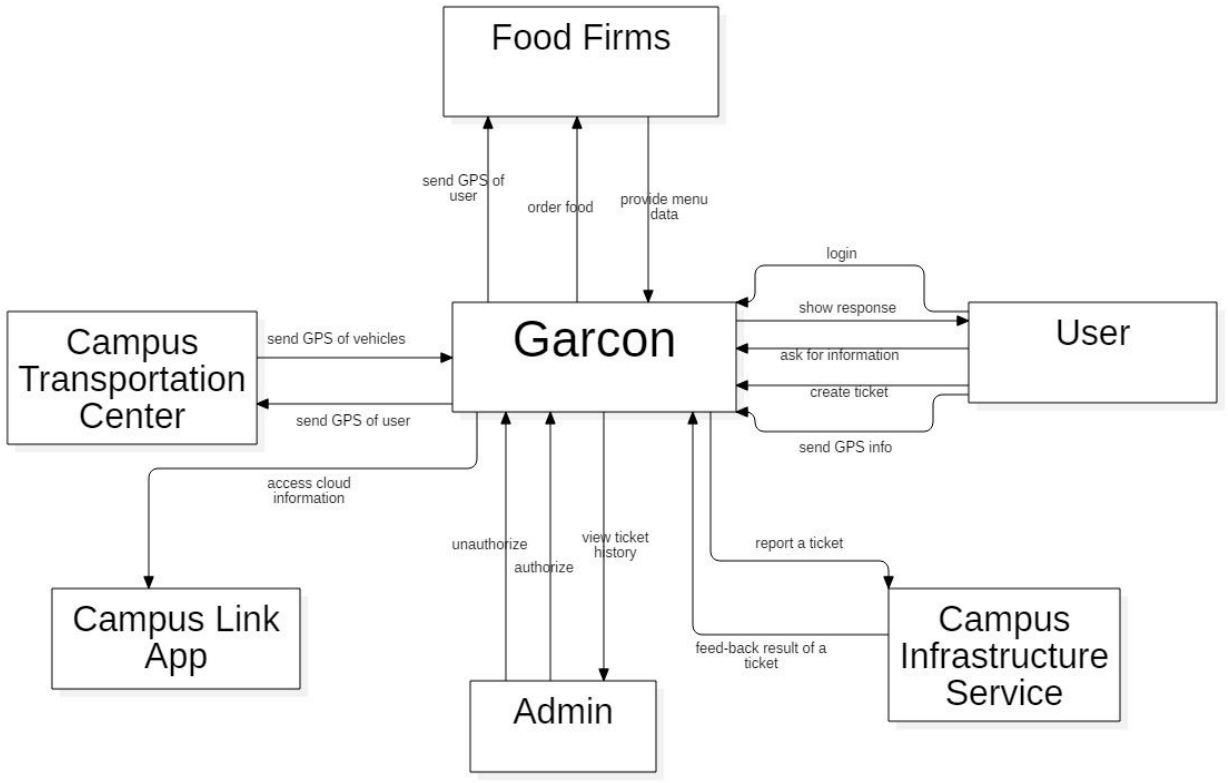

Figure 1: Context Diagram

1.3.1.1. System Interfaces

Infrastructure Service Interface: Since Garcon provides its users the ability to create tickets, there needs to be someone who validate tickets that stored on the cloud and forward the tickets to related department of workers. This is not the case for emergency tickets. In other words, if Garcon gets an alarm from one of sensors in a building, it has an access to directly call the support.

Admin Interface: The system is autonomous itself; however, just for in case of an unexpected problem with system or with the third party firms, system admins are allowed to see all of the history. Admins are also responsible to report any anomaly about requests or reports.

1.3.1.2. User Interfaces

Campus Link Application Entrance Interface: Users will be able to access Campus Link Application through this interface. It basically needs users' ID and password information which was assigned to users during the official campus registration to log in.

Campus Link Application Information System Interface: The application will be a mobile based information application. This is a way that users can interact with Cloud. Through this interface, users will be enabled to see the history and responses of their tickets.

Campus Link Application Food Service Interface: This interface will be showing the information that the food firms in the campus provided. This information will be stored on Cloud and through this interface, users will reach Cloud, not the food firms itself. In case of an order, there will be a choice to share GPS information of user with the food firm until the order reaches to the user.

Campus Link Application Transportation Interface: This interface shows current location of rings and their estimated arrival times.

Campus Link Application Taxi Call Interface: This interface is designed for users to call a taxi. In case of a taxi call, the user will be enabled to see current location of taxi.

1.3.1.3. Third Party Interfaces

Taxi Drivers Interface: In case of a taxi call, an alarm will be sent to driver's interface and a navigation path will be created from taxi driver's current location to the caller's GPS information.

Food Firms Interface: In case of a food order, an alarm will be sent to firm workers' interface and if enabled a navigation path will be created from worker's current location to the GPS information of user who ordered the food.

1.3.1.4. Hardware Interfaces

Wall Embedded Conversation Device: This interface gets input as voice and then translates it into text. After the translation, it sends the text into the system and wait for output. Logging in with their badges, users can create a ticket or ask for any information related to campus using this interface. To deal with urgent situations, on every floor of every building in the campus, this interface will be put.

1.3.2. System Functions

Function	Summary
Login	Users will login to the system via identity and password if they want to use mobile application. If they want to use the devices in building, then they need to show their badges.
Show response	Users will be getting the output of their tickets and information requests both from mobile application and wall embedded conversation device.
Ask for information	Gets information requests that can be made both from mobile application and wall embedded hardware.
Create ticket	Allows users to create tickets related to buildings or any IT staff, then saves these tickets into the system.
Send GPS information	Takes current GPS information of the user.
Report a ticket	Assigns the tickets which are delivered to system into related campus staff.
Feed-back result of the ticket	Allows Campus Infrastructure staff to enter result of a ticket. The result is then saved.
Access all history	Allows system admins to reach all of the requests and tickets. In case of an anomaly, admins are responsible to report.
Authorize	Allows system admins to give access for guests to use the system.
Unauthorize	Allows system admins to block some users temporarily or permanently
Access cloud information	Allows Campus Link application to reach user specific history of tickets, requests and information system.
Send GPS of the user to others	Sends users' GPS information to taxi drivers and food firms in case of a use of these utilities.
Send GPS of vehicles	Sends taxi and ring's current GPS information to the user through mobile application.
Order food	Sends food orders from users to the related food firms.
Provide menu data	Allows food firms to upload and update daily menu and price information.

Table 1: System functions

1.3.3. User Characteristics

There are four different types of users which can be counted as normal users, admins, Campus Infrastructure staff and third parties. Users will reach the system from a mobile

application and wall embedded hardware device and will be able to create tickets and requests to the system. Since the users are generally pursuing an undergrad or grad degree, they are thought as daily technology users. System admins will need basic computer knowledge and need to be able to detect anomalies. Campus Infrastructure staff should be able to check their accounts to deal with tickets and reply the results. So, basic use of computers will be sufficient. Third parties should be able to use their mobile phones as fast as possible to serve fast. For taxi drivers, there is almost nothing to do since the system automatically initiate the navigation to user. However, food firms should be able to correctly upload information about their menu.

1.3.4. Limitations

Regulatory policies: The system is only fully open to admins. However, user names and IDs should be encrypted in order to remove the potential of secret information sharing. So, admins should see the issues and history but should not know the owner of them.

Hardware limitations: Since system aims to serve important functions to a huge number of users in real time, system should be powerful. In addition, wall embedded device should have a sufficient GPU power to convert speech to text.

Interfaces to other applications: System provides all of its applications itself.

Parallel operation: As long as system gets and delivers information nearly in real time, parallelization or serialization does not matter.

Audit and control functions: Since the system accepts third party data which may not always be trusted, admins should be able to delete or block some posts. For instance, food firms may upload completely irrelevant photo instead of food.

Higher-order language requirements: System should be developed with the use of object oriented programming paradigm supported languages since it is a brand new technology and it should be easy to rapidly add new features.

Signal handshake protocols: Top robust and fast protocols should be used to conduct the communication between mobile application interfaces, cloud and wall embedded devices.

Quality requirements: One of the most important requirements of the Garcon is the system being real-time. Since the users are looking for an information or notifying a problem or ordering something, it is the top priority for the system to be fast. Otherwise, system will not satisfy its users.

Criticality of the application: Garcon system should be alive all the day and night since one of its functionalities is to create a ticket and in case of emergence send it immediately to related staff. Thinking of a fire or any disaster, fastest intervention is a must.

Safety and security considerations: Since all of the data is stored on the cloud, it is super important for the cloud to be kept safe. Since the amount of the data to be managed will increase with time, there will be a threshold amount of time for ticket and request history. After the threshold, past ticket and requests will be removed.

Physical/mental considerations: The system can be utilized by all members of a campus since it is designed to be very handy. Also, since the users can not interact with each other on the system, it is no path to disorders like social media addictiveness.

1.4. Definitions

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 2: Definitions

2. References

Alghamdi A., Shetty S. (2016). Survey Toward a Smart Campus Using the Internet of Things. *Conference: 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud). DOI: 10.1109/FiCloud.2016.41*

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

3. Specific Requirements

3.1. External Interfaces

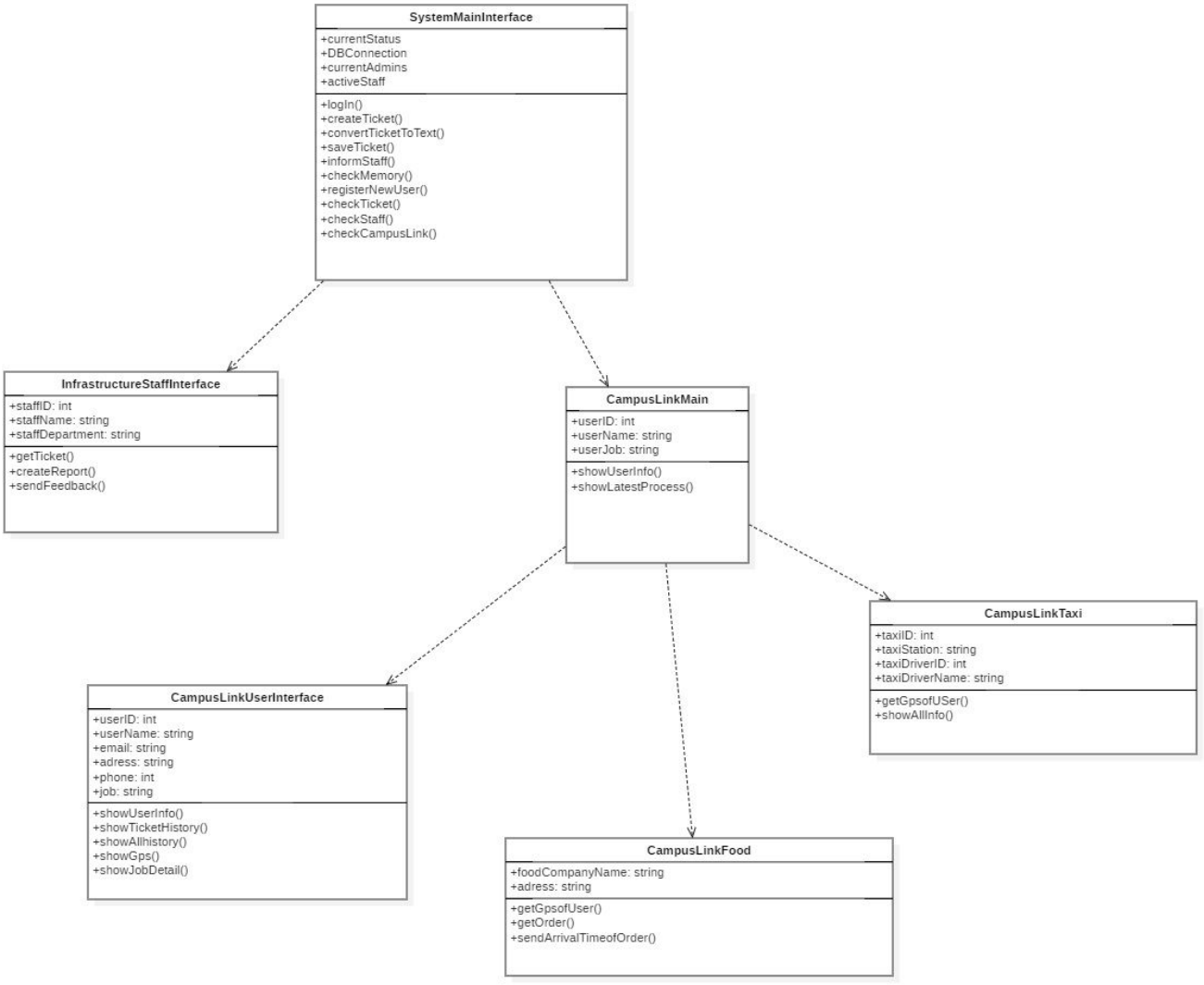

Figure 2: External Interfaces Class Diagram

End-User Communication Interface

req01: User shall log in using his badge with this interface.
req02: User will be asked for his/her requests by this interface.
req03: If user wants to create a ticket, then this interface shall create ticket and convert it to text, after all send it to cloud.
req04: If user wants to call a taxi, this interface shall do this by triggering a request to CampusLink interface.
req05: If user wants to give an order, this interface shall either give information about the food firm or send the request of order to the desired food firm.

Staff Management Interface

req06: The tickets created shall appear to infrastructure service staff in this interface.
req07: Staff shall give feedback about the issue.
req08: Staff shall see the exact place in this interface.

Admin Management Interface

req09: This interface shall show information about users.
req10: Using this interface, admins shall see ticket histories and general system properties.
req11: Admins should be able to authorize/unauthorize using this interface.

Campus Link User Interface

req12: In this interface user shall log in.
req13: User should be able to see their ticket histories, and feedbacks of them .
req14: User should be able to see when a ring will arrive or when his/her taxi will arrive.

Campus Link Taxi Interface

req15: Taxi driver shall see if any request has occured.
req16: Taxi driver shall give response to user and indicate tentative arrival time.

Campus Link Food Interface

req17: Food companies shall send the cloud system their menus and adress information.

req18: Using this interface, food company is notified for order and the company shall indicate the tentative arrival time

3.2. Functions

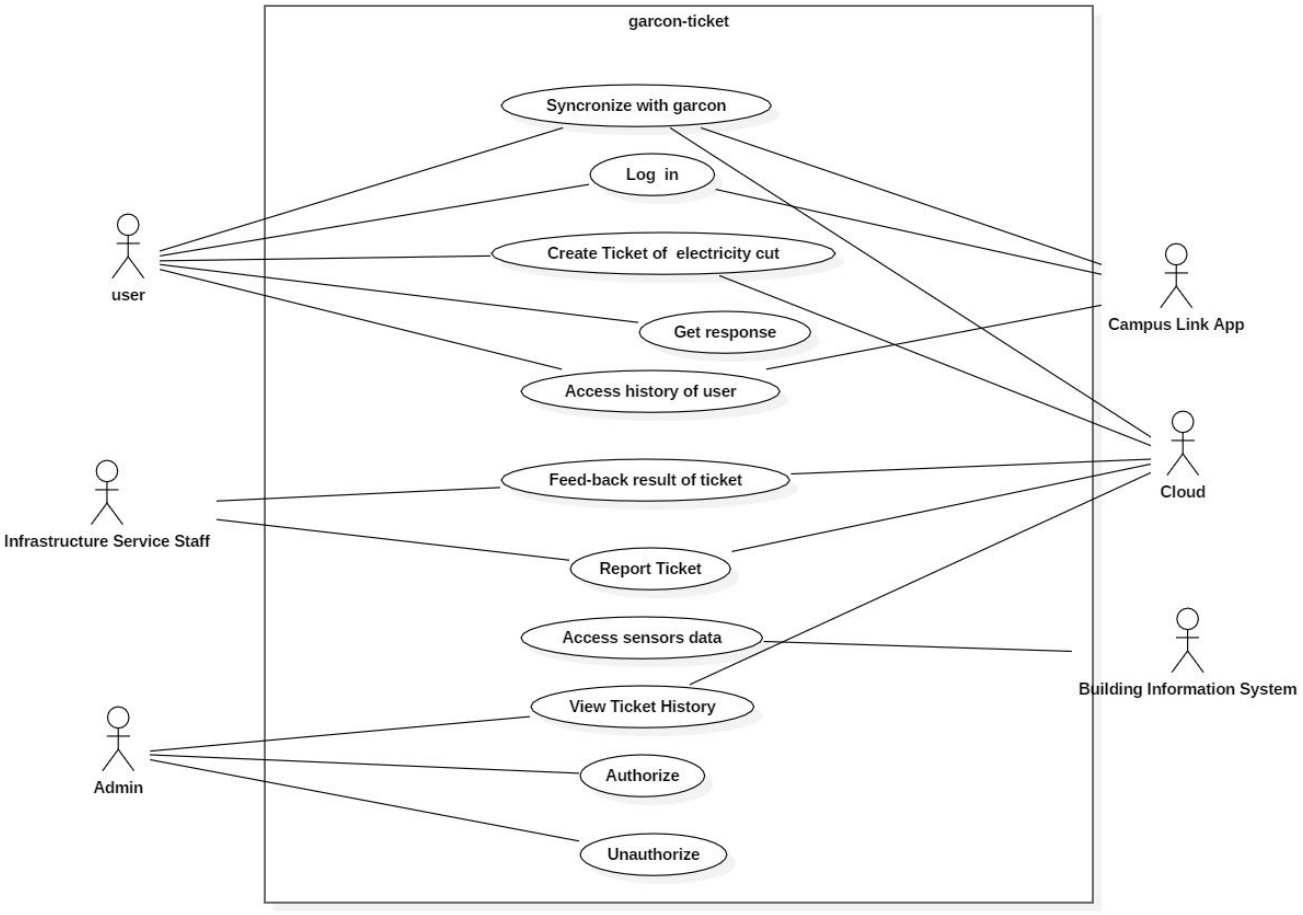

Figure 3: Use Case Diagram for Ticket System

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 3: Create Ticket of Electricity Cut function

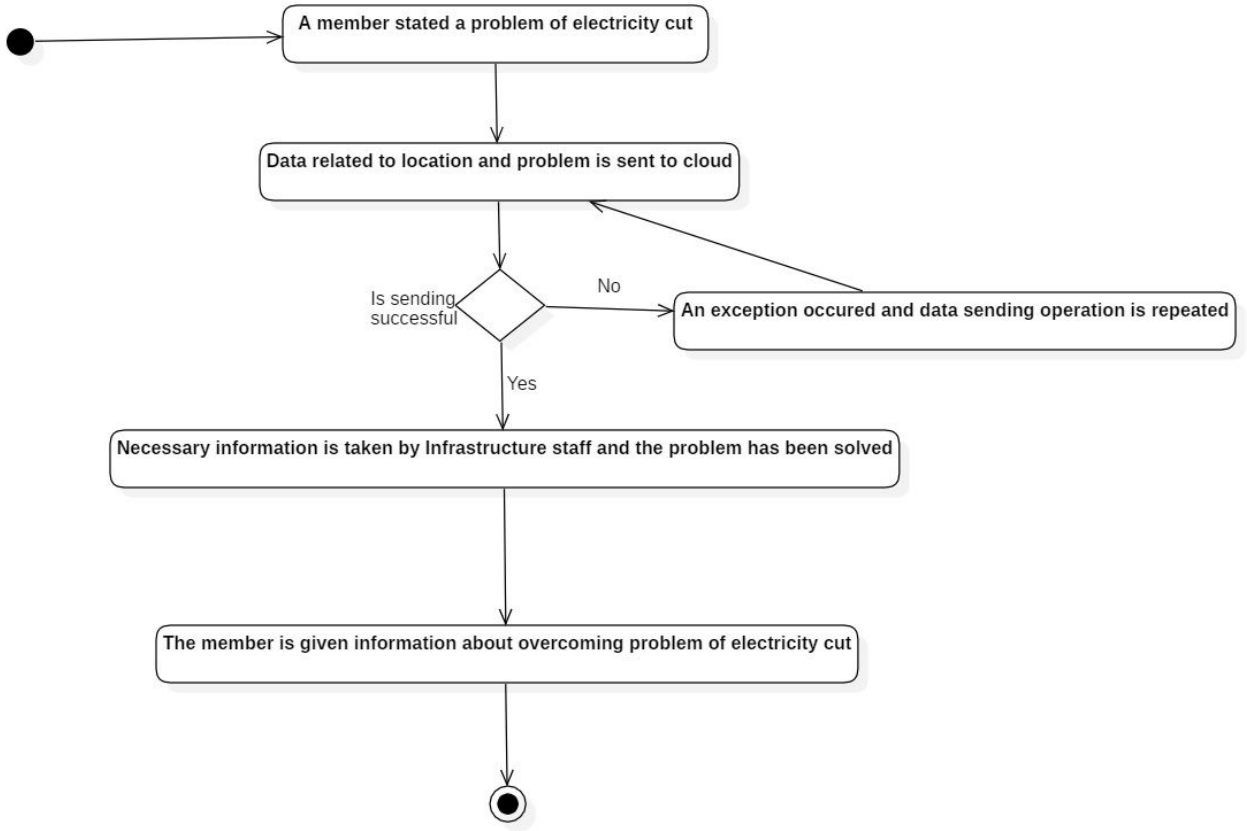

Figure 4: Activity Diagram for Create Ticket of Electricity Cut function

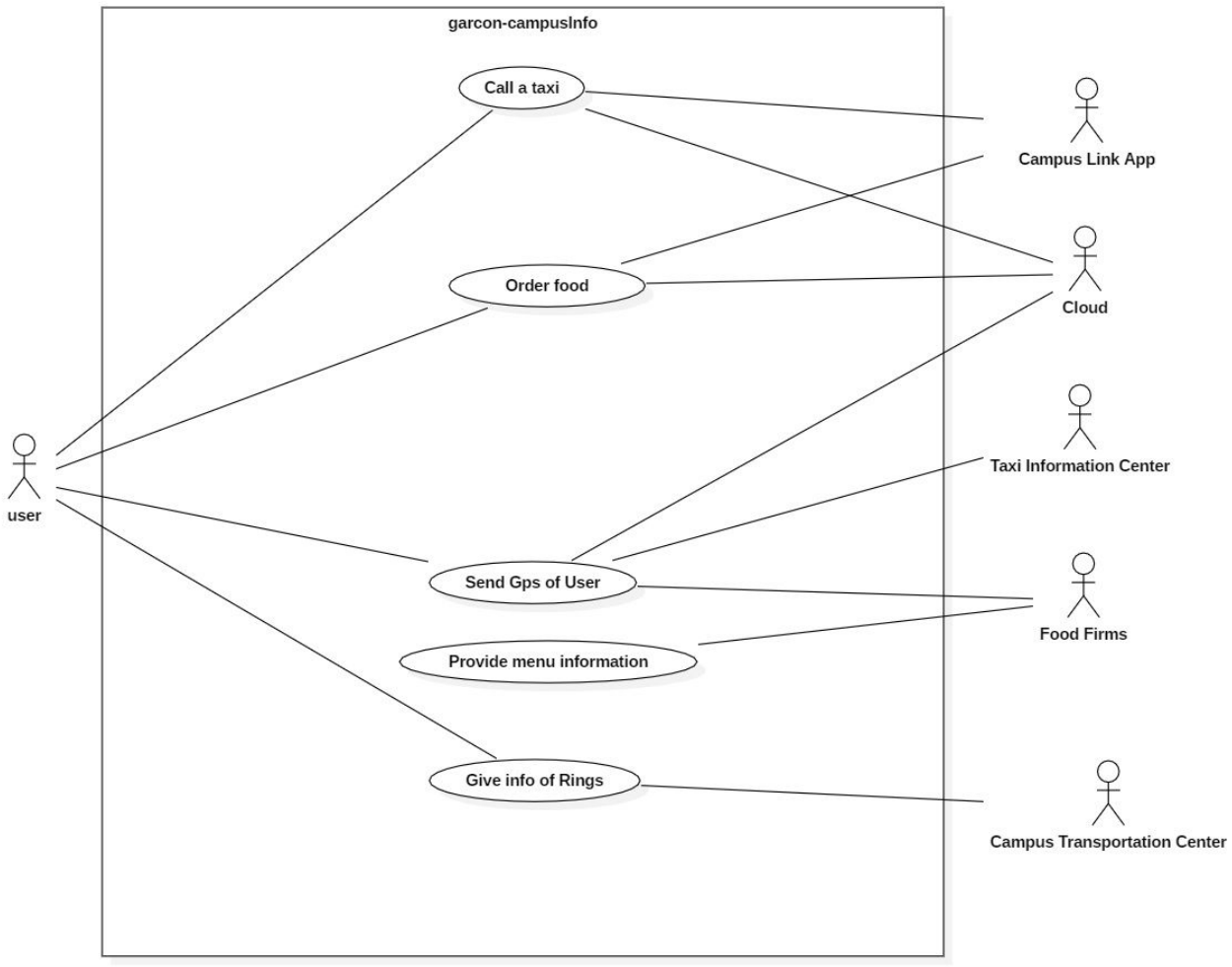

Figure 5: Use case model for campus information system

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 4: Call a Taxi Function

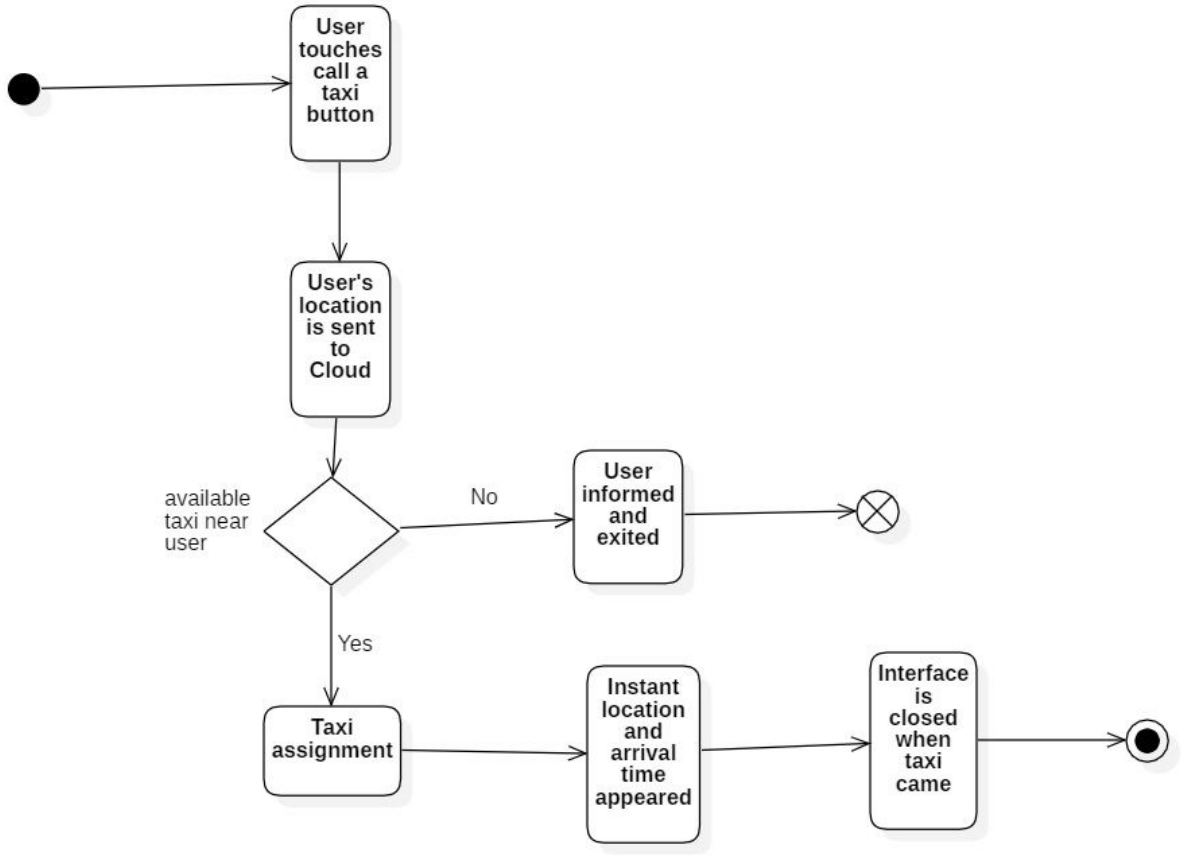

Figure 6: Activity diagram for call a taxi function

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 5: Order Food Function

3.3. Usability Requirements

req19: User shall log in when there is an issue.

req20: The system shall give response at any time that user tries to talk to it.

req21: If a ticket has been created, the system shall warn related staff.

req22: If user does any process by system, user should be able to see that process and his/her log in history in link app.

req23: System admins shall reach any component of system at any time.

req24: If a ticket-reported issue is solved, then user shall be informed.

req25: Both food firms and taxi stations shall see gps of user.

3.4. Performance Requirements

req26: The system shall respond all requests of different users simultaneously.

req27: In a case of ticket creation, the system shall inform staff at the back-end in 1 second.
req28: After staff related to issue solved problem, he shall give information about the current situation in 1 minute.
req29: If user wants to give order from a food company or he wants the system to call a taxi, the system shall inform food company/taxi station in 500 ms.
req30: The system shall give information of exact arrival time of rings.
req31: All processes of the system related to user shall be reported to link app of user.

3.5. Logical Database Requirements

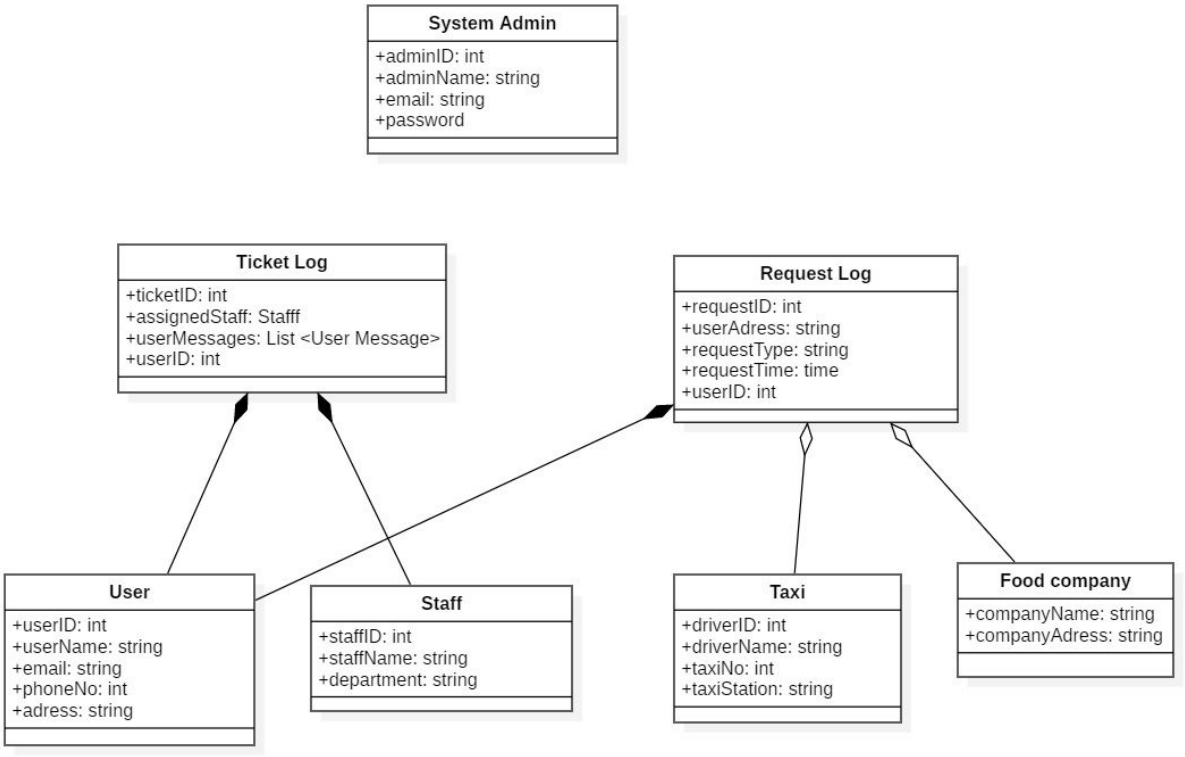

Figure 7: Logical database diagram

req32: Accessibility of all tables shall only be available for system admins.
req33: System admins shall be able to add some entries.
req34: System admins shall add new user.
req35: Entries to request log shall be updated when user make a request.

req36: Request log shall be available for taxi and food company indirectly.
req37: Ticket log shall be updated when user create a ticket.
req38: Ticket log shall be available for staff and user indirectly.

3.6. Design Constraints

req39: For storing the data of user, the regular standards of Metu shall be applied.
req40: The system shall keep information related to user in accordance with priority laws.

3.7. Software System Attributes

Reliability
req41: The system admin shall test reliability in terms of time and proper response of the system per week.
req42: Synchronization between the system and link app shall be safe.
req43: To prevent collisions between tickets and requests of multiple users, the system shall have substitute cloud system.

Availability
req44: The system shall be available all the time, except for some development or repairing purposes it shall be offline.

Security
req45: Login and other private information of users shall be open to system admins only.
req46: The connection between the system and infrastructure service staff or the connection between the system and other third-party systems shall be safe against attacks.

Maintainability
req47: Documentation shall be easily understandable, readable, updatable.
req48: Any update or development process shall be performed in accordance with international standards.

Portability
req49: Campus link app shall be available for different OS.
req50: The system shall have multiple natural language options.

req51: The core code behind the system shall be convertible any other platform.

3.8. Supporting Information

4. Verification

5. Appendices

5.1. Assumptions and dependencies

5.2. Acronyms and abbreviations