Garcon

Software Design

Description

Ramazan Selim Şahin 2171999

Gürkan Kısaoğlu 2171726

**Table of Contents**

[1. Introduction 5](#_Toc6077298)

[1.1 Purpose of the System 5](#_Toc6077299)

[1.2 Scope 5](#_Toc6077300)

[1.3 Stakeholders and Their Concerns 5](#_Toc6077301)

[2. References 6](#_Toc6077302)

[3. Glossary 7](#_Toc6077303)

[4. Architectural Views 8](#_Toc6077304)

[4.1 Context View 8](#_Toc6077305)

[4.2 Composition View 8](#_Toc6077306)

[4.3 Information View 10](#_Toc6077307)

[4.3.1 Interfaces 10](#_Toc6077308)

[4.3.2 CRUD Operations 10](#_Toc6077309)

[4.4 Interface View 10](#_Toc6077310)

[4.4.1 Internal Interfaces 10](#_Toc6077311)

[4.4.2 External Interfaces 10](#_Toc6077312)

**List of Figures**

Figure 1: Component Diagram..............................................................................8

Figure 2: Deployment Diagram.............................................................................9

**List of Tables**

# Introduction

## Purpose of the System

## Scope

## Stakeholders and Their Concerns

**User:** Usersare only University students, who can access the system with their ID Cards. Their primary concern is getting campus information no matter where they are and giving feedbacks to relevant persons via opening issues to make campus a better place.

**Worker:** Workers are University employees who can access the system with their ID Cards just like Users. Their concern is solving issues opened by the Users.

**Admin:** Admins are the people who can access the system via a web interface. Their primary concern is manage the system users(students, workers).

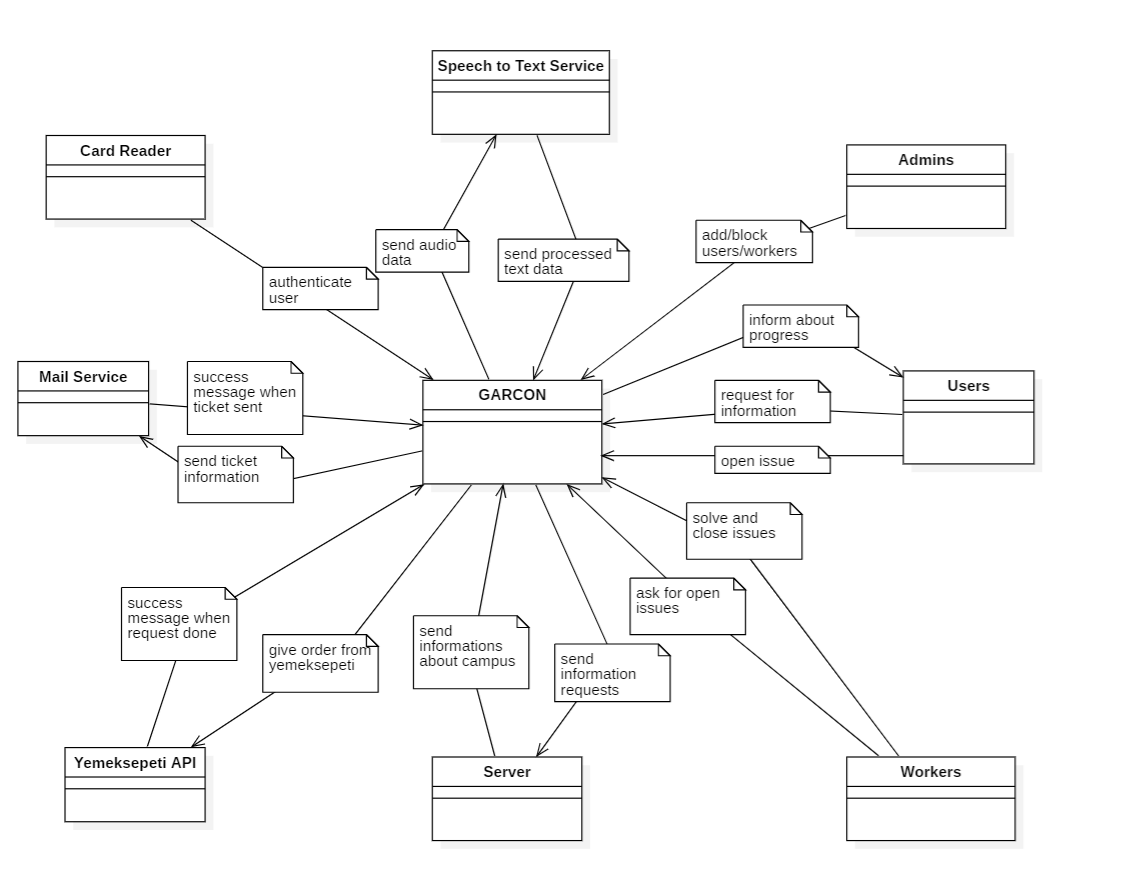
# References

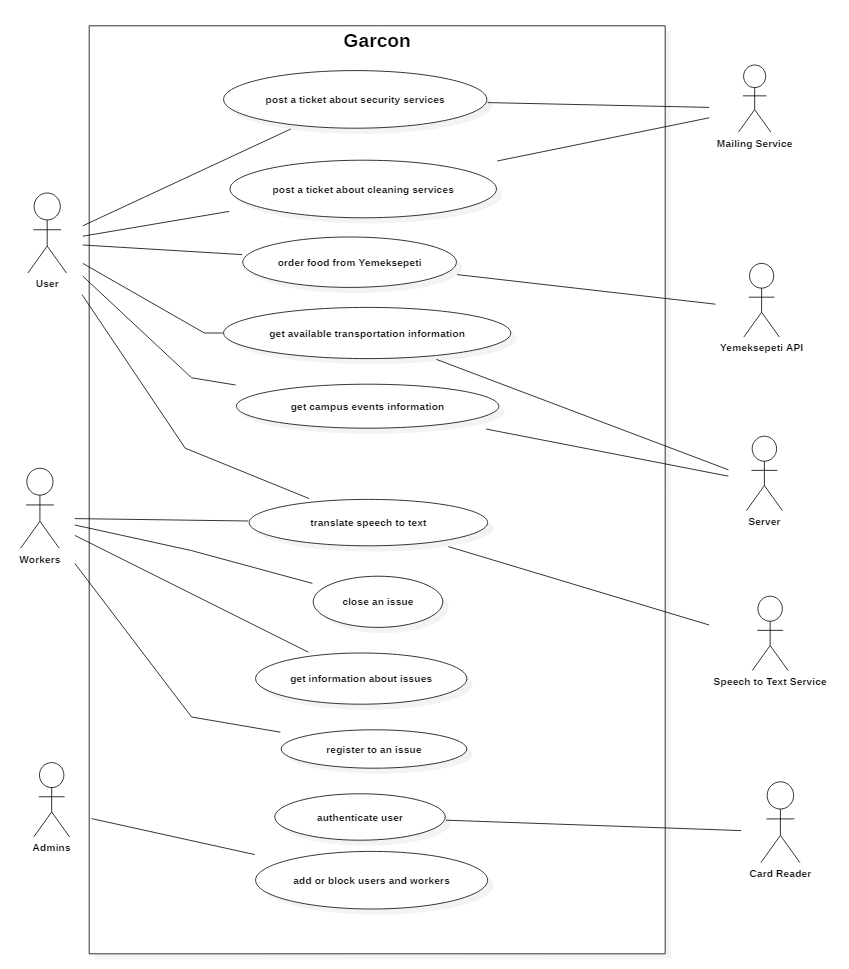
# Glossary

# Architectural Views

## Context View

In this viewpoint, all system use cases and descriptions explained with details. These descriptions show how system should behave in specific situations and how the functionalities implemented. Context diagram shows how users and other subsystems interacts with each other and use case diagram and functionality descriptions specified below the context diagram.





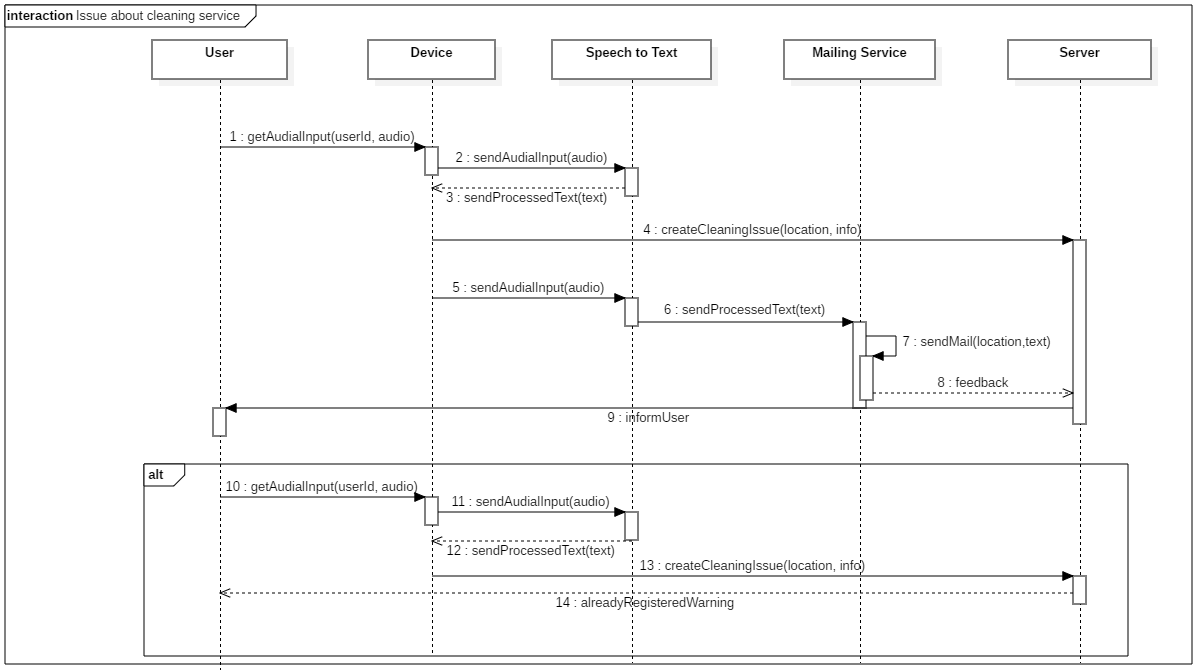
**Figure 3: Use Case Diagram**

|  |  |
| --- | --- |
| **Use case name** | Issue about security service |
| **Actors** | Users, Speech to Text Service, Mailing Service, Server |
| **Description** | If a user notices a security issue he/she can notify related workers via Garcon. |
| **Data** | Audial input from user |
| **Preconditions** | User should be authenticated |
| **Stimulus** | User giving audial input about issuing a security request |
| **Basic Flow** | Step 1 – User gives audial input  Step 2 – Audial input gets processed by Speech to Text Service  Step 3 – Issue is created and server side is informed  Step 4 – Processed text of audial input is sent as an email to the related workers  Step 5 – User is informed that issue is registered |
| **Alternative Flow** | Step 3 – System detects same request already issued  Step 4 – Importance level of request is updated  Step 5 – User is informed that issue is already registered |
| **Exception Flow** | - |
| **Post conditions** | An issue instance is created on system and related workers are informed. |

Table 1: Open security issue function

|  |  |
| --- | --- |
| **Use case name** | Issue about cleaning service |
| **Actors** | Users, Speech to Text Service, Mailing Service, Server |
| **Description** | If a user notices a cleaning issue he/she can notify related workers via Garcon. |
| **Data** | Audial input from user |
| **Preconditions** | User should be authenticated |
| **Stimulus** | User giving audial input about issuing a cleaning request |
| **Basic Flow** | Step 1 – User gives audial input  Step 2 – Audial input gets processed by Speech to Text Service  Step 3 – Issue is created and server side is informed  Step 4 – Processed text of audial input is sent as an email to the related workers  Step 5 – User is informed that issue is registered |
| **Alternative Flow** | Step 3 – System detects same request already issued  Step 4 – Importance level of request is updated  Step 5 – User is informed that issue is already registered |
| **Exception Flow** | - |
| **Post conditions** | An issue instance is created on system and related workers are informed. |

Table 2: Open cleaning issue function



**Figure 4: Issue about cleaning service Sequence Diagram**

|  |  |
| --- | --- |
| **Use case name** | Order food from Yemeksepeti |
| **Actors** | Users, Speech to Text Service, Yemeksepeti API |
| **Description** | When user asks for ordering food, the request translated into text first, gets analyzed and then system automatically give an order from Yemeksepeti |
| **Data** | Audial input from user |
| **Preconditions** | User should be authenticated |
| **Stimulus** | User giving audial input about ordering food |
| **Basic Flow** | Step 1 – User gives audial input  Step 2 – Audial input gets processed by Speech to Text Service  Step 3 – A request is posted to Yemeksepeti Api  Step 4 – Success message shown to the user |
| **Alternative Flow** | - |
| **Exception Flow** | If any error occurs or restaurant is closed , system will show a log message. |
| **Post conditions** | User gives an order from Yemeksepeti. |

Table 3: Order food function

|  |  |
| --- | --- |
| **Use case name** | Get transportation information |
| **Actors** | Users, Speech to Text Service, Server |
| **Description** | Information service on transformation info. User can get a possible route to a direction from the device he/she is now interacting with and user can get information about transportation schedules. |
| **Data** | Audial input from user, current transportation services’ data, map data |
| **Preconditions** | Worker should be authenticated |
| **Stimulus** | User giving audial input about getting informed on transportation |
| **Basic Flow** | Step 1 – User gives audial input asking directions  Step 2 – Audial input gets processed by Speech to Text Service  Step 3 – Best possible route and its transformation information is obtained from the server  Step 4 – Information is converted to audio form by Speech to Text Service  Step 5 – User is informed |
| **Alternative Flow** | Step 1 – User gives audial input asking transportation schedules  Step 2 – Audial input gets processed by Speech to Text Service  Step 3 –Transformation information on ring, bus, subway services is obtained from the server  Step 4 – Information is converted to audio form by Speech to Text Service  Step 5 – User is informed |
| **Exception Flow** | - |
| **Post conditions** | User is informed with best routes and transportation information and information about this query is saved to database to inform further queries faster. |

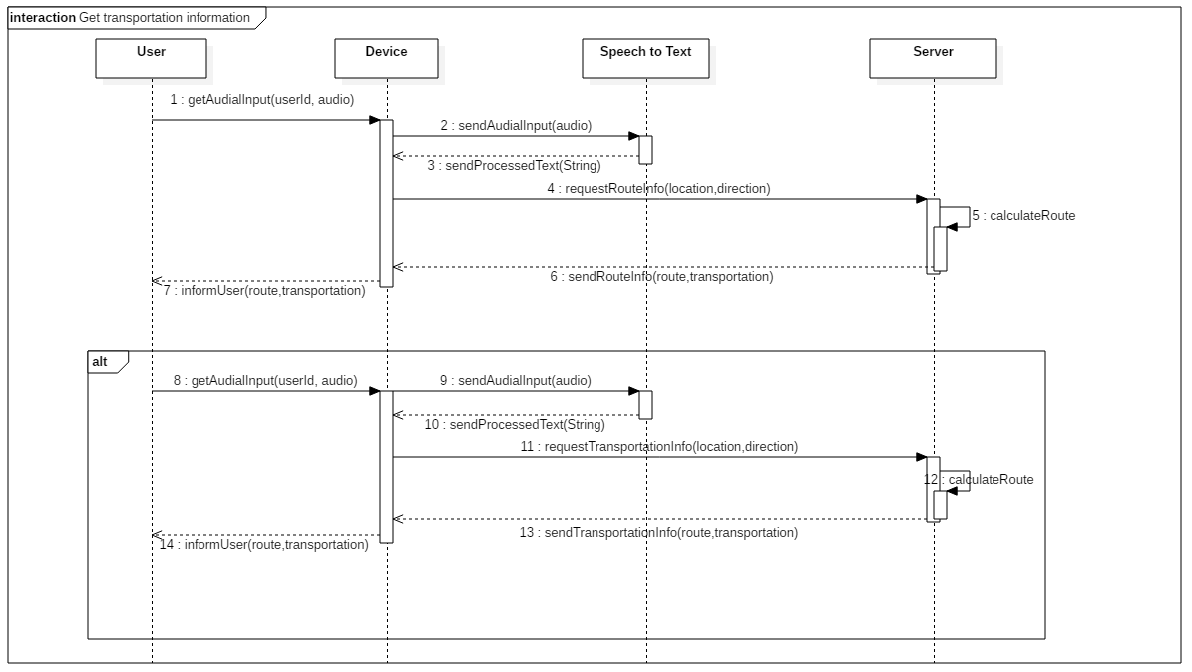


Table 4: Get transportation info function

**Figure 5: Get transportation information Sequence Diagram**

|  |  |
| --- | --- |
| **Use case name** | Get campus events informations |
| **Actors** | Users, Speech to Text Service, Server |
| **Description** | When user asks for available events in campus, server interacts the database and submits events for user’s information. |
| **Data** | Audial input from user |
| **Preconditions** | User should be authenticated |
| **Stimulus** | User giving audial input about campus event informations |
| **Basic Flow** | Step 1 – User gives audial input  Step 2 – Audial input gets processed by Speech to Text Service  Step 3 – Available events searched in server  Step 4 – Available events are converted to audio format.  Step 5– Events listed to the user. |
| **Alternative Flow** | Step 4 – If no event is available on campus, Garcon will not give any listings. |
| **Exception Flow** | If any error occurs, system will show an error message |
| **Post conditions** | System shows all events |

Table 5: Get event information function

|  |  |
| --- | --- |
| **Use case name** | Close an issue |
| **Actors** | Workers, Speech to Text Service, Mailing Service, Server |
| **Description** | Workers close issues they have handled. |
| **Data** | Audial input from worker |
| **Preconditions** | Worker should be authenticated |
| **Stimulus** | Worker giving audial input about closing an issue |
| **Basic Flow** | Step 1 – Worker gives audial input  Step 2 – Audial input gets processed by Speech to Text Service  Step 3 – Issue is closed and server side is informed  Step 4 – Worker is informed that issue is closed |
| **Alternative Flow** | - |
| **Exception Flow** | - |
| **Post conditions** | The issue is marked as solved on system. |

Table 6: Close issue function

|  |  |
| --- | --- |
| **Use case name** | Register to an issue |
| **Actors** | Workers, Speech to Text Service, Server |
| **Description** | Workers register to an issue to prevent possible conflicts. |
| **Data** | Audial input from worker, current registered workers on the issue |
| **Preconditions** | Worker should be authenticated |
| **Stimulus** | Worker giving audial input about registering to an issue |
| **Basic Flow** | Step 1 – Worker gives audial input  Step 2 – Audial input gets processed by Speech to Text Service  Step 3 – Current workers on issue are displayed to worker  Step 4 – Worker is asked a confirmation after seeing current workers on issue  Step 5 – With workers confirmation he is registered to issue (Database update) |
| **Alternative Flow** | - |
| **Exception Flow** | If worker does not confirm after seeing current workers on the issue the process is aborted. |
| **Post conditions** | The worker is registered to issue. |

Table 7: Register issue function

|  |  |
| --- | --- |
| **Use case name** | Getting informations about issues |
| **Actors** | Workers, Speech to Text Service |
| **Description** | Worker scans his/her id card and Garcon gets activated. Then waits for worker to talk to decide what to do. |
| **Data** | Audial input from worker |
| **Preconditions** | Worker should be authenticated |
| **Stimulus** | Worker giving audial input about open issues |
| **Basic Flow** | Step 1 – Worker gives audial input  Step 2 – Audial input gets processed by Speech to Text Service  Step 3 – Server returns open issues from Database |
| **Alternative Flow** | - |
| **Exception Flow** | - |
| **Post conditions** | Worker can see the whole available/open issues from database. |

Table 8: Get issue information function

|  |  |
| --- | --- |
| **Use case name** | Authentication |
| **Actors** | Users, Workers, Admins, Card Reader |
| **Description** |  |
| **Data** | Chip from Id Cards |
| **Preconditions** | - |
| **Stimulus** | Id Card must be scanned |
| **Basic Flow** | Step 1 – User/Worker/Admin holds Id Card to the device  Step 2 – Device reads the Card and authenticate |
| **Alternative Flow** | - |
| **Exception Flow** | Step 1 – Card doesn’t recognized or cannot be scanned  Step 2 – System displays a visual error output |
| **Post conditions** | Authenticate successful and device gets waiting for audial input to work. |

Table 9: Authentication

|  |  |
| --- | --- |
| **Use case name** | Adding users |
| **Actors** | Admin |
| **Description** | Admin adds new user to database |
| **Data** | - |
| **Preconditions** | Admin should be authenticated |
| **Stimulus** | Admin giving information about new user |
| **Basic Flow** | Step 1 – Admin gives information about user  Step 2 – Input gets processed by Speech to Text Service  Step 3 – User Id Card Scanned  Step 4 – New user added |
| **Alternative Flow** | - |
| **Exception Flow** | Step 1 – Admin gives information about user  Step 2 – Input cant be recognized  Step 3 – System gives a warning |
| **Post conditions** | New user added to the database. |

Table 10: Add user function

|  |  |
| --- | --- |
| **Use case name** | Blocking users |
| **Actors** | Admin |
| **Description** | Admin blocks user from database |
| **Data** | - |
| **Preconditions** | Admin should be authenticated |
| **Stimulus** | Admin gives delete command and gives information |
| **Basic Flow** | Step 1 – Admin gives information about user to be deleted  Step 2 – Input gets processed by Speech to Text Service  Step 3 – User deleted |
| **Alternative Flow** | - |
| **Exception Flow** | Step 1 – Admin gives information about user  Step 2 – Input cant be recognized  Step 3 – System gives a warning |
| **Post conditions** | User deleted from the database. |

Table 11: Block user function

## Composition View

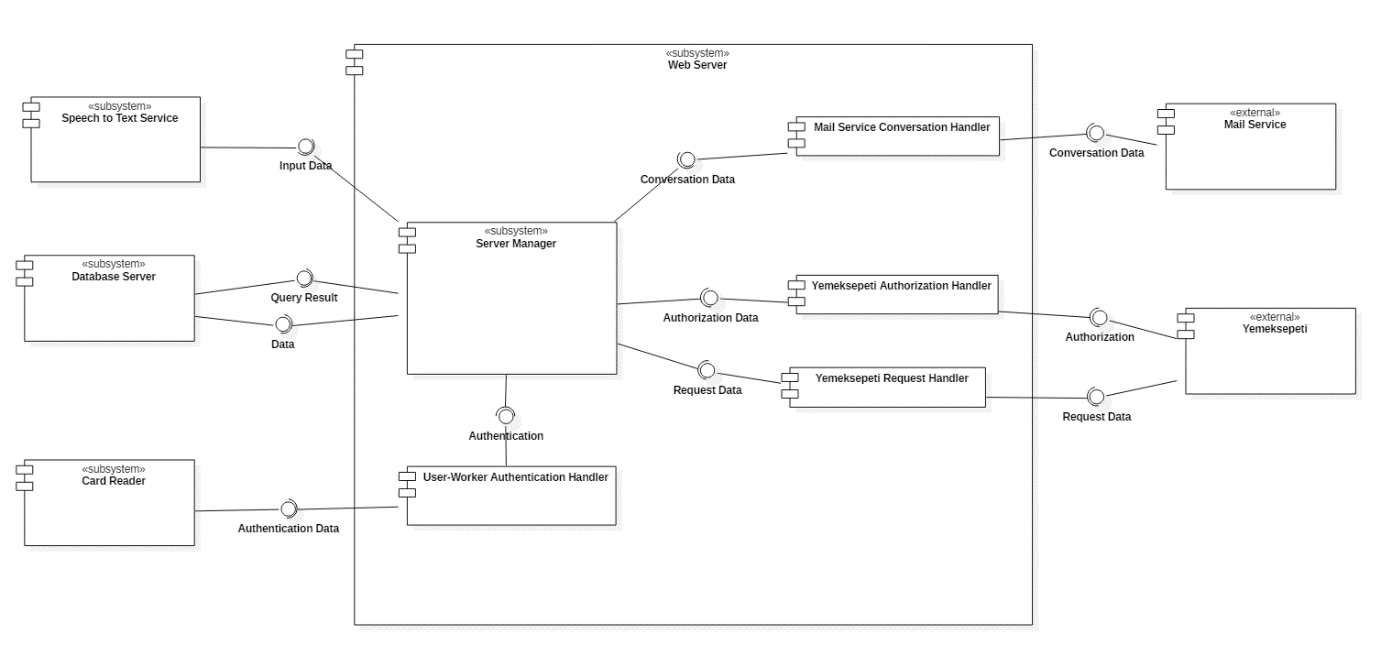


Figure 1: Component Diagram

**Design Rationale:**

* Server Manager is the fundamental component, which manages/handles information flow and data flow in the Web Server. Moreover, it authorizes/authenticates users and workers.
* Card reader is the component for authentication of users and workers. When Card Reader reads data from ID card, it will direct Authentication Data to User-Worker Authentication handler.
* User-Worker Authentication handler component is responsible from user and worker authentication to the system. It processes raw authentication data coming from card reader.
* Speech to Text Service is the main input provider for system. It converts given audial data to processable data that will be forwarded to Server Manager.
* Database Server is the component that securely stores the related data and provides it when needed.
* Since mailing is the main conversation between workers and system itself, Mail Service is an external component, which is dedicated for managing conversation.
* Mail Service Conversation Handler is the component responsible for providing processed conversations and related data (timestamps, target workers…).
* Yemeksepeti is the external component for ordering food. Two components are responsible from the interaction between Server Manager and Yemeksepeti. Firstly, Yemeksepeti Authorization Handler manages the authorization of user to Yemeksepeti. Secondly, Yemeksepeti Request Handler processes and manages users’ orders from Yemeksepeti.

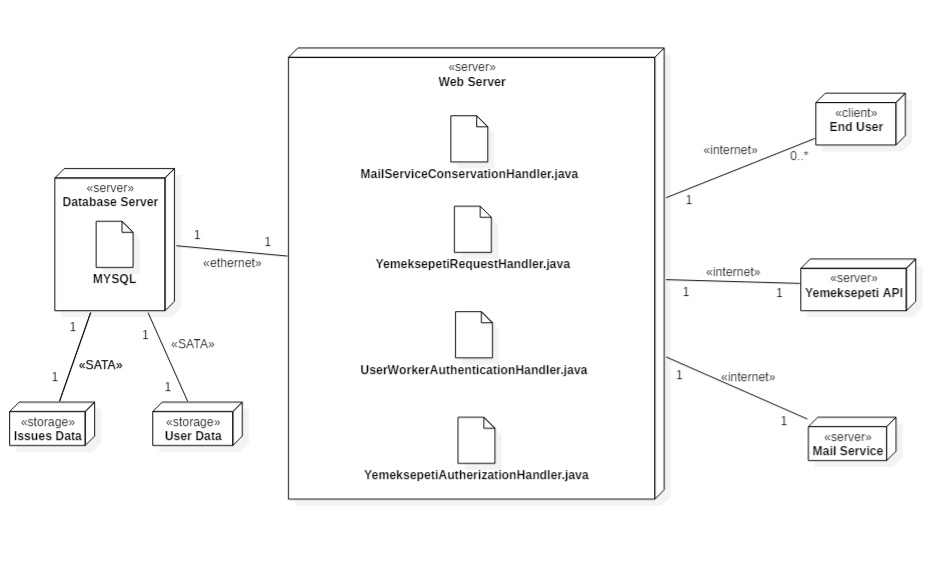


Figure 2: Deployment Diagram

**Design Rationale:**

* We use Java/Spring in web server side and DB is managed by MySQL.
* The end users who uses Garcon to get information or opening an issue will communicate with system with talking.
* Garcon will take audial input and translate into a text input and sends appropriate handler.
* There are two separate db storage:
  + Issue storage is for storing all issues.
  + User storage is for storing all information about admins/workers/regular users.
* We SATA for all storage since it has much larger storage space to store whole issue/user.
* All internet based connections will use encrypted communication protocols.

## Information View

### Interfaces

### CRUD Operations

## Interface View

### Internal Interfaces

### External Interfaces