Garcon

Software Requirements

Specification

Ramazan Selim Şahin 2171999

Gürkan Kısaoğlu 2171726

# Table of Contents

**1.Introduction5**

1.1 Purpose of the System5

1.2 Scope5

1.3 System Overview5

1.3.1 System Perspective5

1.3.2 System Functions6

1.3.3 User Characteristics6

1.3.4 Limitations6

1.4 Definitions6

**2. References6**

**3. Specific Requirements6**

3.1 External Interfaces6

3.2 Functions7

3.3 Usability Requirements19

3.4 Performance Requirements19

3.5 Logical Database Requirements19

3.6 Design Constraints19

3.7 Software System Attributes19

3.8 Supporting Information19

# Table of Figures

Figure 1: Context Diagram.....................................................................................6

Figure 2: Use Case Diagram..................................................................................7

# Table of Tables

Table 1: Open Security Issue Function.................................................................8

Table 2: Open Cleaning Issue Function................................................................9

Table 3: Order Food Function………………………………………………………..10

Table 4: Get Transportation Info Function...........................................................11

Table 5: Get Event Information Function……………………………………………12

Table 6: Close Issue Function…..........................................................................13

Table 7: Register Issue Function.........................................................................14

Table 8: Get Issue Information Function..............................................................15

Table 9: Authentication………………………………………………………………..16

Table 10: Add User Function.................................................................................17

Table 11: Block User Function...............................................................................18

# Introduction

This document is Software Requirement Specification for Microsoft’s smart campus project, called Garcon.

## Purpose of the System

The purpose of the project is making METU campus smarter than ever. Making the campus an interactive environment for both students and workers. When there is a campus wide security issue, or environment issue students can immediately inform workers of campus with this system. Also, it enables students to get information about campus transportation and food possibilities.

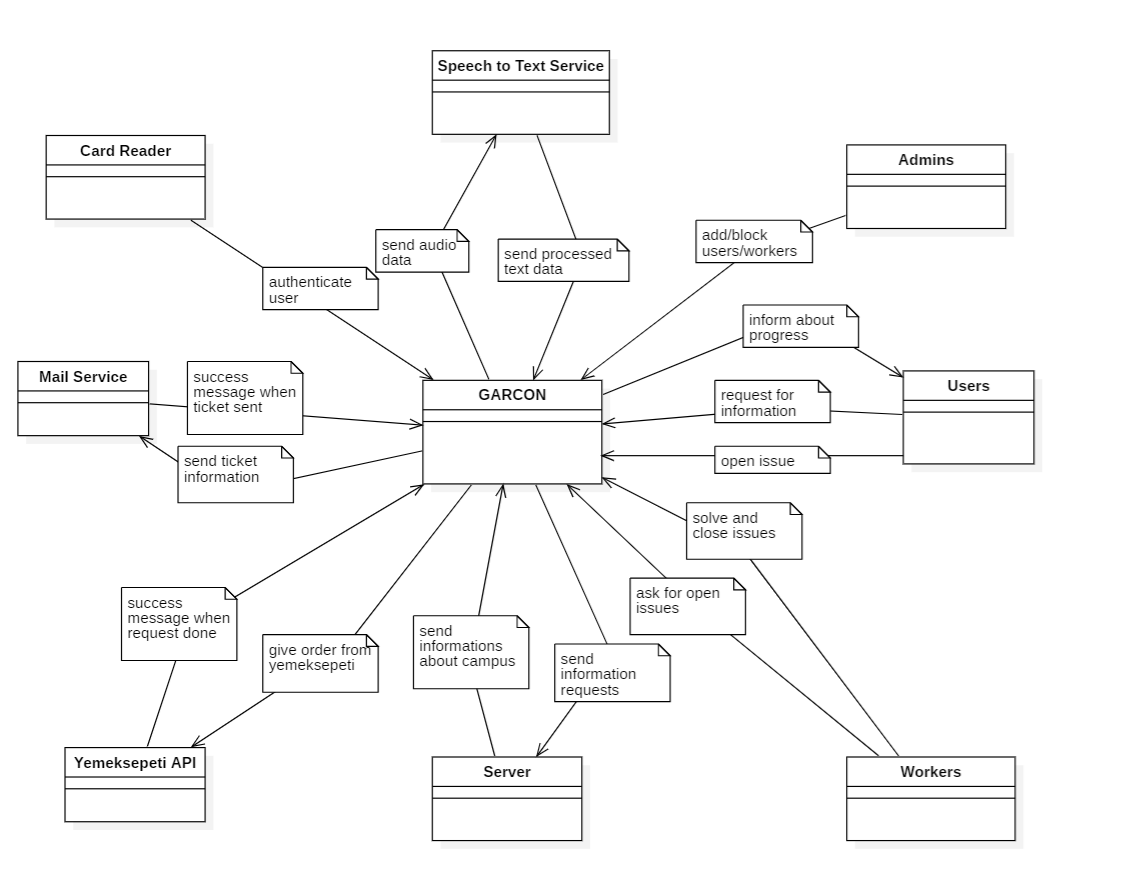
## 1.2 Scope

## 1.3 System Overview

This section will give general information about the system.

### 1.3.1 System Perspective

General purpose of this system is making students life in campus much easier than before. System uses a card reader device for student id cards to authenticate students. Then, waits for the student to talk. When student talks, speech to text service analyses the speech and decides whether student opening an issue or asking for an information. After this stage, Garcon will do whatever students want automatically. If an issue opened , mailing service activated; or if an information asked, then Garcon will get the information to the student from various services.



**Figure 1: Context Diagram**

### 1.3.2 System Functions

### 1.3.3 User Characteristics

### 1.3.4 Limitations

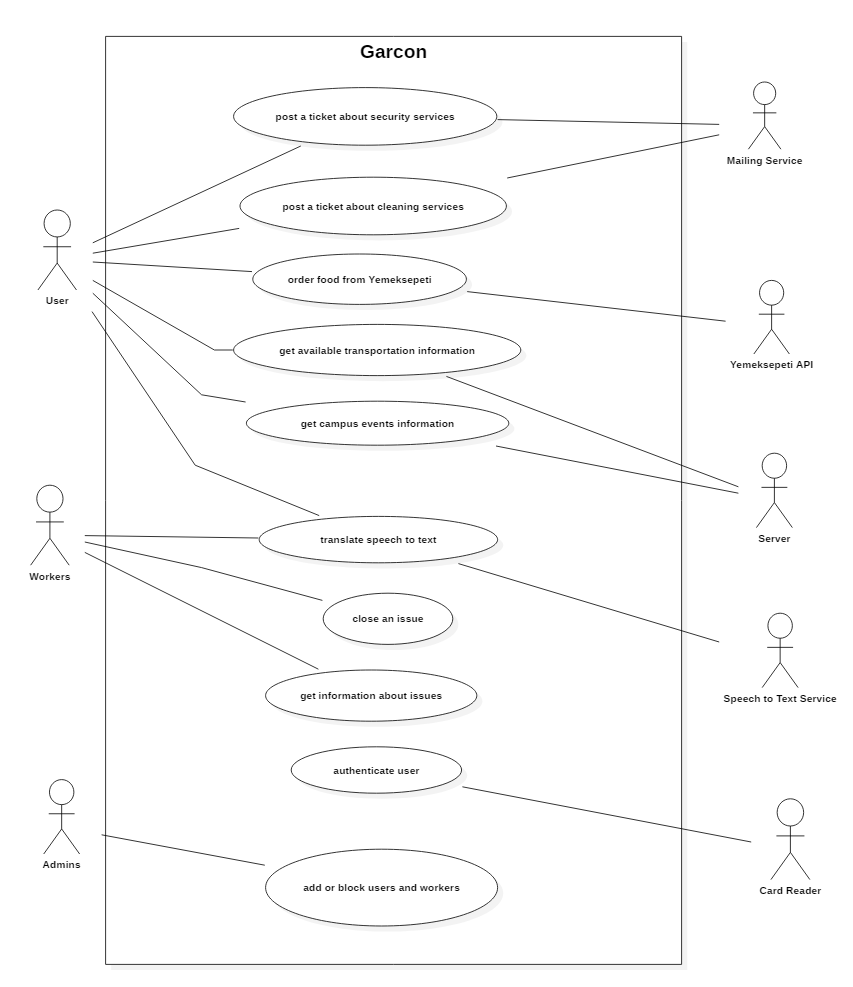
## 1.4 Definitions

# References

# Specific Requirements

## 3.1 External Interfaces

## 3.2 Functions



**Figure 2: 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

|  |  |
| --- | --- |
| **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

|  |  |
| --- | --- |
| **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

## 3.3 Usability Requirements

## 3.4 Performance Requirements

## 3.5 Logical Database Requirements

## 3.6 Design Constraints

## 3.7 Software System Attributes

## 3.8 Supporting Information