Unique Queue Application (Süleyman Gölbol | Arash Ziaee)

Software Requirements Specifications (SRS)

1. Presentation of the problem

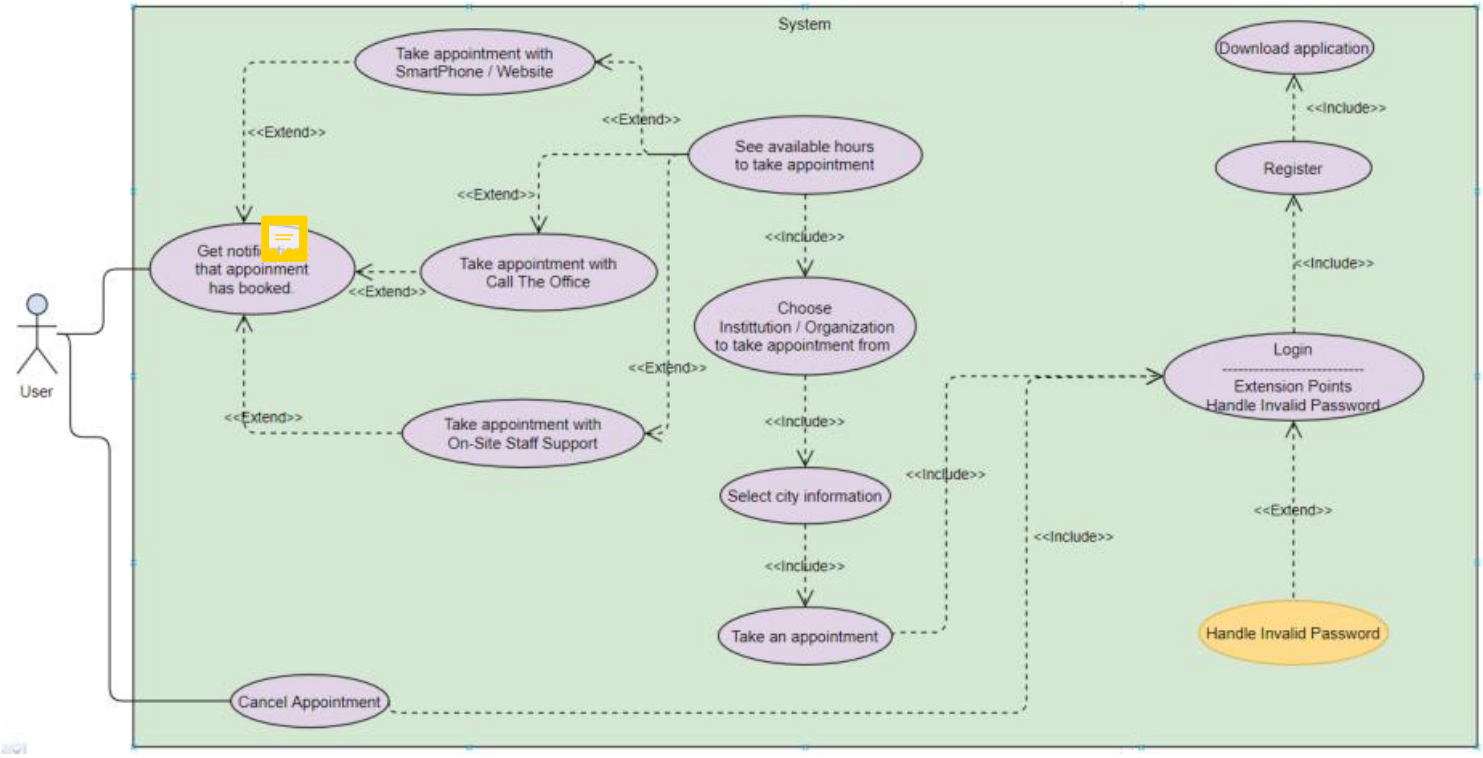
What is happening if every reservation/appointment gets done in one ambient?
After COVID-19 every municipal organization etc., they are trying to have their website or system for making appointment reservations.
So, imagine a 100-organization, each of them has their way of having an appointment. This is going to be so confusing for most of the people to manage their reservations.

2. Brief presentation of the solution

Our solution is to design an application that collects every essential information (reservation, e-mails, etc...) in a user-friendly environment. So, every user can easily have access and it is not going to be so confusing for people.

3. Use cases

•3a – Use Case diagram



- **3b – Use case descriptions**

Süleyman Gölbol - Use-Case Description

• Name: Take Appointment with Smartphone/Website

• Actor: User

• Requirements:

- Users – Need to take appointment from an institution.

• Trigger: User clicks the take an appointment button.

• Basic Flow:

1- User selects city information.
2- User selects institution.
3- User selects available hours.
4- User takes appointment with default method.
5- User get informed about appointment with notification.
6- Use case ends.

• Postconditions:

The user can only take appointment if there is available hour that he/she wants.

The user can only cancel the appointment until 24 hours before.

• Alternative flows:

3a: If that hour is not available:

3a1. User have to find an other hour for appointment.

3a2. Use case ends.

4a: User wants to take appointment with smartphone/website.

4a1. User takes appointment with smartphone/website.

4a2. User gets informed about appointment or cancels and use case ends.

4b: User wants to take appointment with staff support.

4b1. User takes appointment with staff support.

4b2. User gets informed about appointment or cancels and use case ends.

4c: User wants to take appointment with calling the office.

4c1. User takes appointment with calling the office.

4c2. User gets informed about appointment or cancels and use case ends.

Arash Ziaee - Use case description

- A user wants to make an appointment for office work on a certain day and time.
- Need to log In.
- Need to select the city and office.
- Select the Service that he/she needs.
- Select the date and time to reserve the appointment.
- Reserve the Appointment.
- Name: Take The appointment with Smartphone/Website
- Actor: The User
- Requirements:

The User - Needs to make an appointment remotely and at a certain time and day.

Information - The user needs to specify for which operation he is making an appointment.

- Trigger: The need of doing some office work and book appointments as soon as possible.
- Basic flow:

1- Open the application or Website on his phone

2- Select the city and Office.

3- Select the service that the User needs for making a reservation.

4- Select the available time and day.

- PostConditions:

The application works just the city and office supported by the application.

The appointment can be done if the time that the user selects is available.

- Alternative flow:

4a: The Application is not showing the timetable

4a1:The application gives the user the other option for making the appointments

4b: The time and the day is not available for selecting

4b1:The application through the server of the office shows the available day and time.

- **3b.2 – Functional and non-functional requirements**

Functional Requirements:

1- User could be able to register with the e-mail address.

2- System should handle password errors.

3- Before choosing city, user cannot choose institution.

4- The appointment can be done if the time that the user selects is available.

5- The user can only cancel the appointment until 24 hours before.

6- The application only works on specific cities.

7- System should be able to connect to database to check or add accounts.

8- Institution should be connected to database to be informed.

Nonfunctional Requirements:

1- The system shouldn't allow informants share information without logging in.

2- The system shouldn't sell personal information of users because of privacy reasons.

3- The system should be packaged as a mobile application.

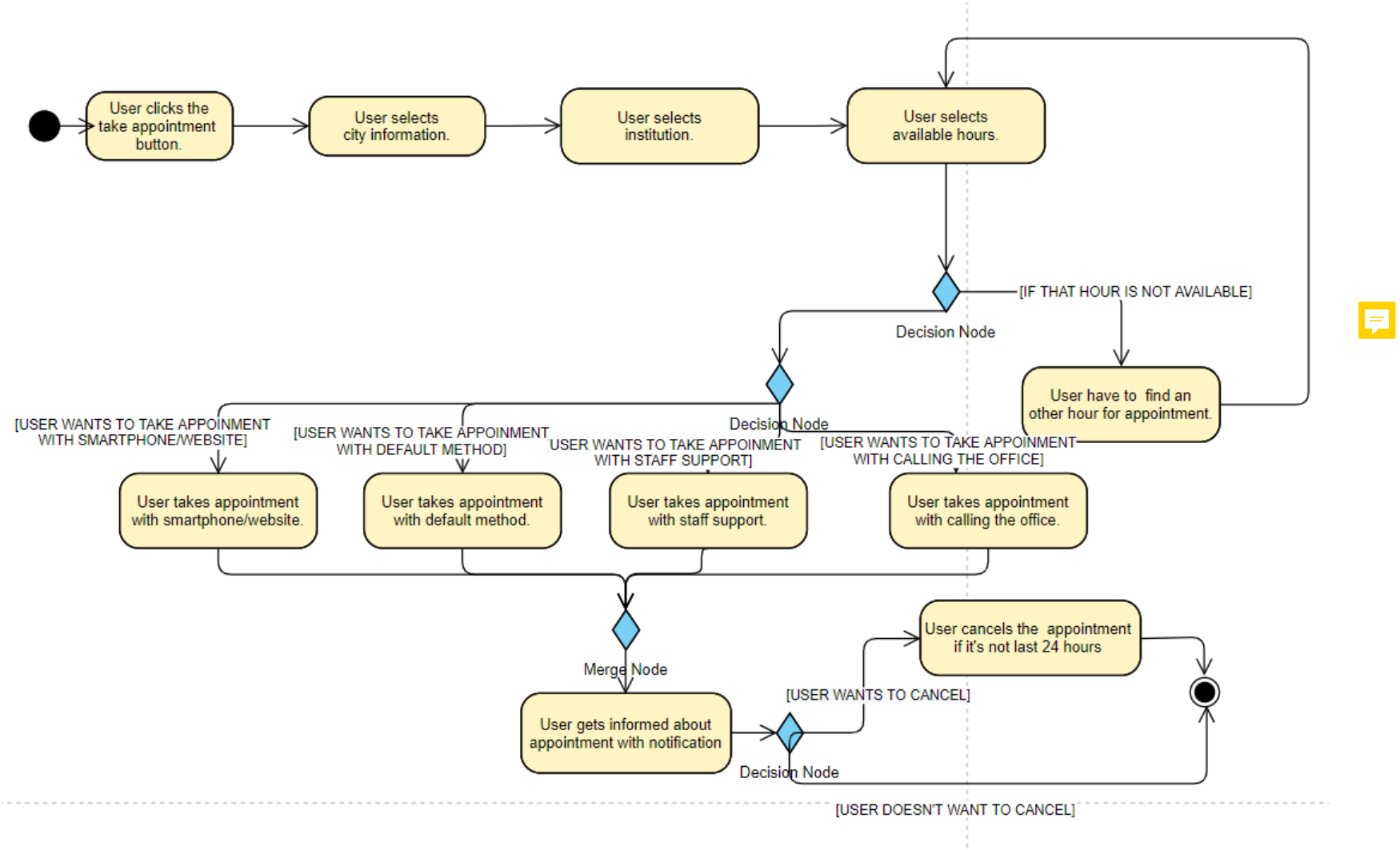
4- The system should have the ability to save user feedback after each reservation.

5- the system should be used by people who want to take appointment.

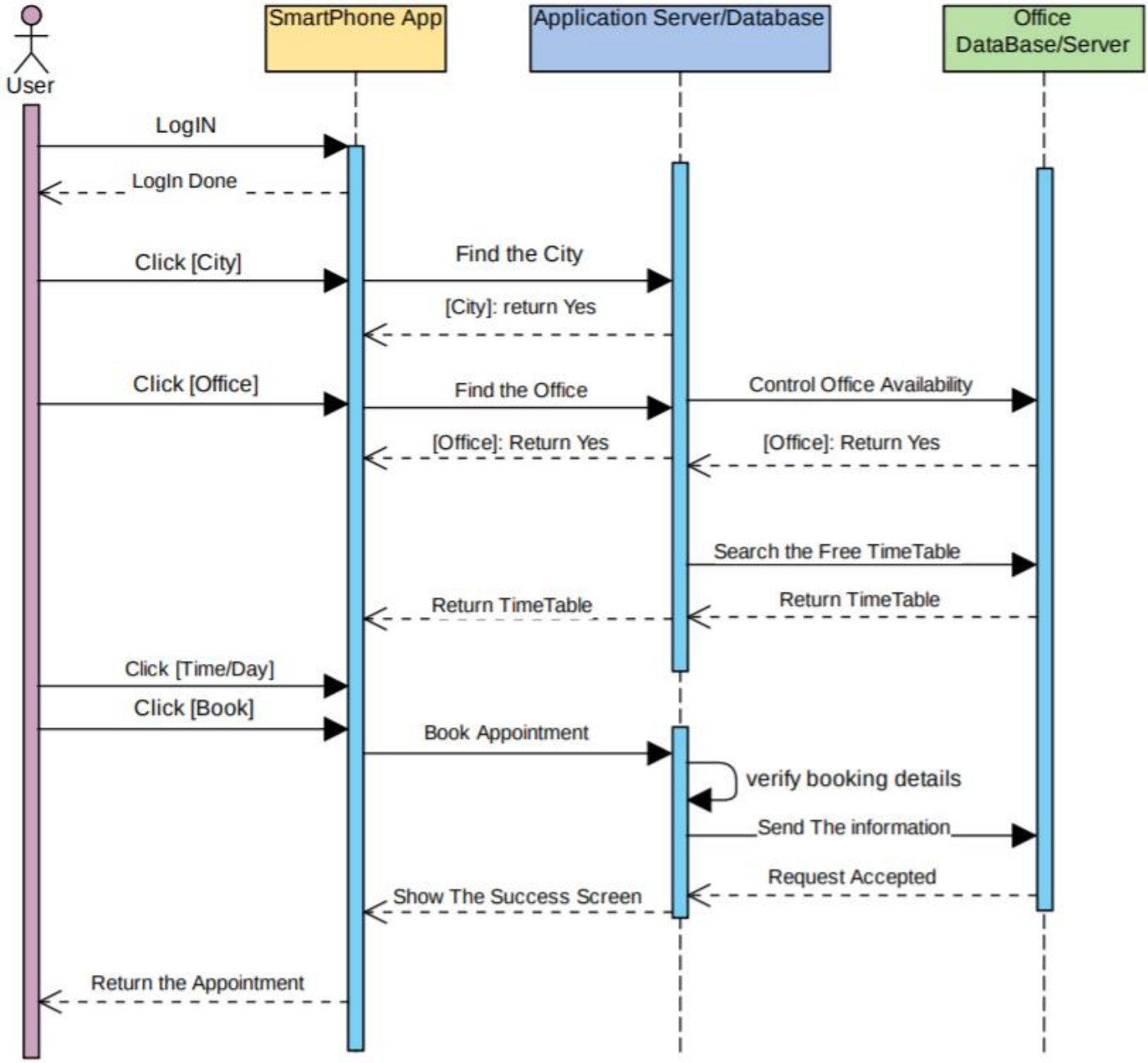
6- The application should be portable.

•**3c - Additional UML diagrams explaining UC**

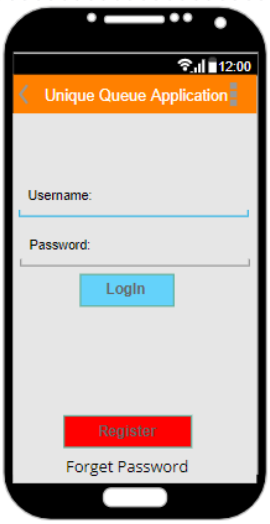
Süleyman Gölbol – Additional Diagram

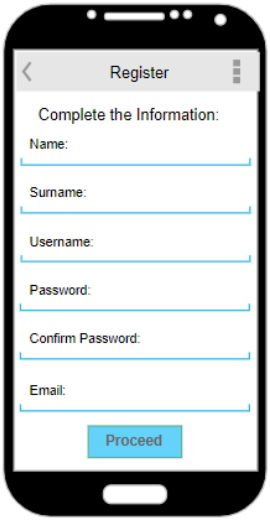


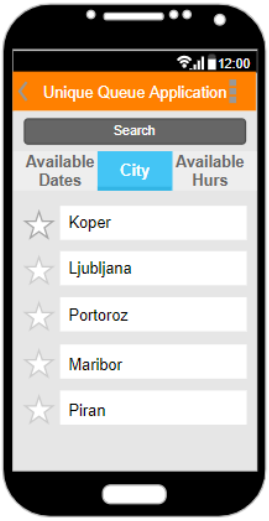
Arash Ziaee – Additional Diagram

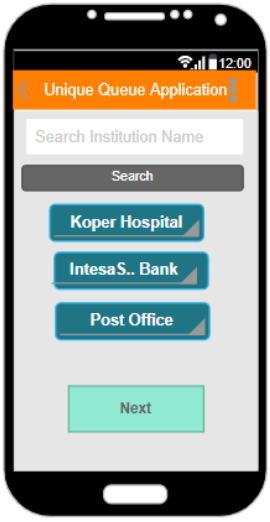


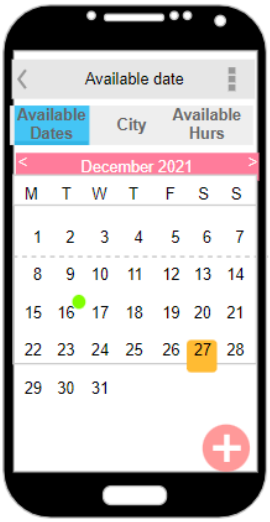
4. GUI - sketch of user interface

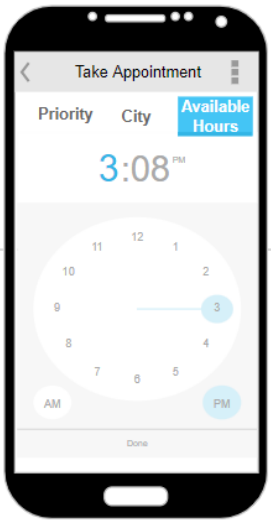


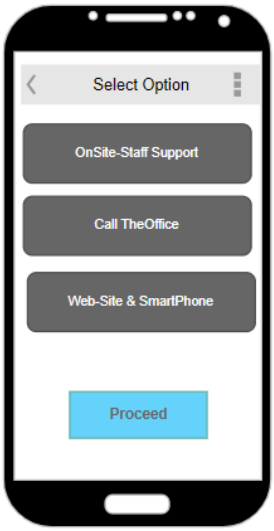


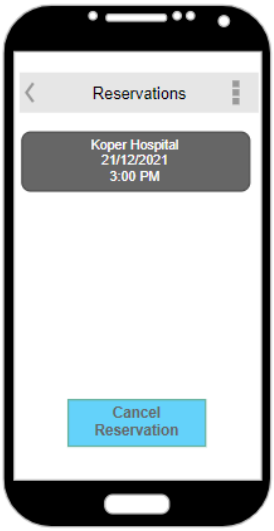












5. Connections to other systems – API

Firebase Database REST API

We are going to use Google Firebase for connect our application to database that will be stored on Microsoft SQL Server 2019 / Google Firebase Realtime Database.

Google Firebase uses REST API. When a client request is made with a REST API(with the other name: RESTful API), it transfers the status of the resource to the requester.

This information is can be sent by JSON, HTML or PHP. REST API uses layered system architecture and helps us with Client-server decoupling.

5. Miscellaneous

If institution doesn't have available hour in that day, it's shouldn't be visible on list.

User cannot choose an institution before choosing city.

User should choose available date from the system.

We need to include pictures, tables, charts of the institutions on the application.

When user cancel the reservation, user should receive an e-mail.