
# DESIGN DOCUMENT

## Software Engineering

Supervisor:
Prof. Matteo Giovanni Rossi

| Authors: | ID: |
| --- | --- |
| Ahmed Ibrahim Abdelrazzak Hamed | 10682755 |
| Khaled Said Ahmed Maamoun | 10696857 |
| Mahmoud Mohamed Aboelwafa Medany | 10715340 |

# Table of contents

# List of Figures

# 1. Purpose of the document

This document provides a specification on the architecture of AYLI. This document discusses the Requirements Analysis and Specification in more details and further description. The main project architecture and design pattern are proposed, and different runtime behaviors are explained using detailed sequence diagrams.

# 2. ARCHITECTURAL DESIGN

## 2.1 Overview

AYLI has three-tier architecture: presentation tier, logic tier, and a data tier. The main benefit of this kind of structural implementation the flexibility to have several database servers holding the data or the integration with cloud system.

- Presentation tier: the entry point of the user interaction with the central system. The web application is considered in this tier. It is used to invoke the main server for storing and retrieving from database. This tier has been built in React Application on web technologies such as HTML, JavaScript, CSS, using ReactJS, Redux, React Router JavaScript libraries, and communicates with the other layers through API calls.

- Logic tier: This layer contains logical operations for performing tasks like processing commands, validation, logical evaluations, decision making, database interactions, and web applications structuring. It communicates between the website and the database. This tier is implemented using JavaScript.

- Data tier: The data tier includes the data persistence mechanisms (database servers, file shares, etc.). This tier is based on Google Firebase API connected to the application tier that exposes methods of managing the stored data without exposing or creating dependencies on the data storage mechanisms. This tier uses Google Firebase API including the Authentication service, Cloud Firestore service, Storage service, Hosting service and Cloud Functions service.

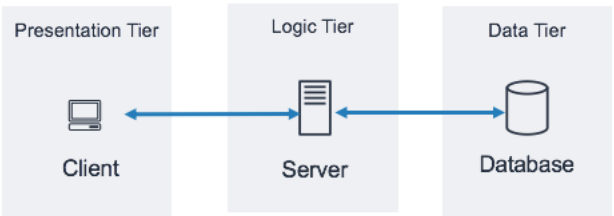


*Figure 1 General System Architecture.*

## 2.2 Component View

### 2.2.1 AYLI Business Logic

This component solves all the tasks related to generation of SOR, proposals, invoices, claims and payment. In addition to, the processes of exchanging information between both users and AYLIs, and the management of registration and authentication tasks.

- Authentication Manager

It manages the accounts, both of USrs and AYLIs. It also handles the signing in and logout phases of them both and the signing up. It exposes the following interfaces: USrs Interface: for the interaction with the USrs. AYLIs Interface: for the interaction with the AYLIs.

- Network Manager

Every communication among the Server and the Clients goes through this component; it routes incoming messages toward the right component and sends outgoing messages at the recipient application client.

- USrs Manager

It handles the functionalities of USrs accounts. Through the presentation level, it provides the right interface to the USrs.

- AYLIs Manager

It handles the functionalities of AYLIs accounts. Through the presentation level, it provides the right interface to the AYLIs.

- Service Manager

This component takes care of all the aspects of the service management. It takes incoming reports from Users and check if they correspond to already notified violations. It uses external Places Service, Geocoding Service and Map Service to obtain the locations of the service centers close to the USrs position. This component handles the generation of SOR as well as the update of its status. Operations of viewing, accepting or rejecting a SOR, carried out by AYLIs, are also under the responsibility of the Service Manager.

- Proposal Manager

After the SOR has been accepted by the AYLIs, then, he/she is able to generate a Proposal for that service. All that concerns the managing of proposals is handled by the Proposal Manager. It allows to generate proposals, send proposals and evaluate proposals, this last one means, to accept or reject a proposal by USrs.

- Invoice Manager

This component takes care of all the aspects of the invoice management. It allows to generate claims, get invoice status and update invoice status.

• Mailing Manager

This component takes care of all the aspects of the mail management. It allows to generate and send messages to both the AYLIs and USrs. It includes messages sending as a consequence of the invocation of other component methods or messages which are sent directly between them.

• Payment Manager

After accepting an invoice, the USrs must pay for the service. This component takes care of all the aspects of the payment management. It allows to insert the data of his/her credit card and pay for the service.

• Review Manager

After a service (or invoice) has been completed by the AYLIs and the device has been returned to the USRs, then, the last one is able to review a Proposal for that service. All that concerns the managing of review is handled by the Review Manager. It allows to rate services and generate comments related to the services.

• Claim Manager

After any completed service the SOR has been able to request claim with that service. It allows to generate claims, send claims and respond claims.

• Buffer Manager

After a service (or invoice) has been completed by the AYLIs, USrs is able to review a Proposal for that service. All that concerns the managing of review is handled by the Review Manager. It allows to rate services and generate comments related to the services.

• Data Store Manager

This component realizes an intermediate layer between the Database and each component that accesses it; it provides to these components an easier way to store and retrieve data from the Database. Data Store Manager create queries and directly interacts with the Database. It is linked to every other component who needs to access the Database.

### 2.2.2 External Interfaces

Both Client Application and Business Logic Layer use functions form external component with which they communicate. We are introducing these components with some more particulars:

• Places API

It is used by AYLI System to retrieve in an intelligent and comprehensive format the USrs address from his/her GPS position.

• Geocoding API

The Geocoding API provides a direct way to access the following services via an HTTP: converting addresses into geographic coordinates, which you can use to place markers on a map, or position the map; converting geographic coordinates into a human-readable address and finding the address for a given place ID.

• Maps Javascript API

It is used by AYLI system to graphically show on a map, the service location around the position of the USrs.

• Database (Google Firebase API)

Where AYLI System store all its data. It is external to the Business Logic Layer, and the Data Store Manager is in charge of making this layer communicate with the Database.

## 2.3. Deployment View

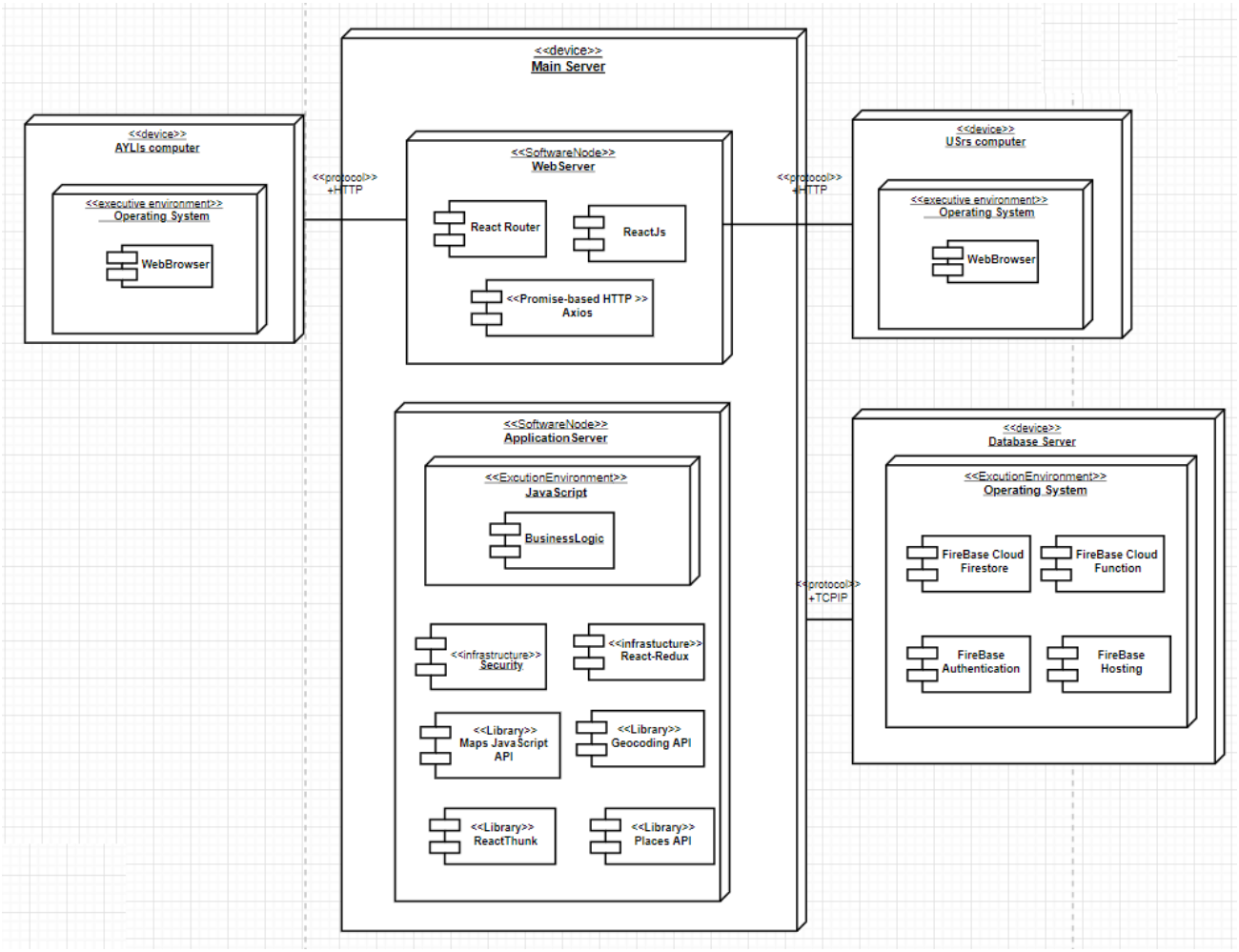


*Figure 2 Deployment diagram*

## 2.4. Runtime View

The main activities are explained in the following sub sections, to describe the interaction of various system component to run these activities.

### 2.4.1. USrs signs up

Components involved and their roles:

• USrs (Application)

the activity starts when the USrs, from the website and submits the signup application, by previously filling out all the fields. When the button “sign up” is pressed, the client calls the method signup of the Authentication Manager component passing as parameters all the data of the form. He then expects as response either an affirmative message with the confirmation of the sign up, or a negative message due to some problems during the process.

• Authentication Manager

once the Authentication Manager has been called by the USrs, first check whether the data inserted is valid in terms of type of data, then it calls the Data Storage Manager method addUser through the Data Store Manage component, which is the linked with the Database.

• Data Store Manager

it invokes a query to the Database, in order to save the information.

• Database

It stores the information of the new user.

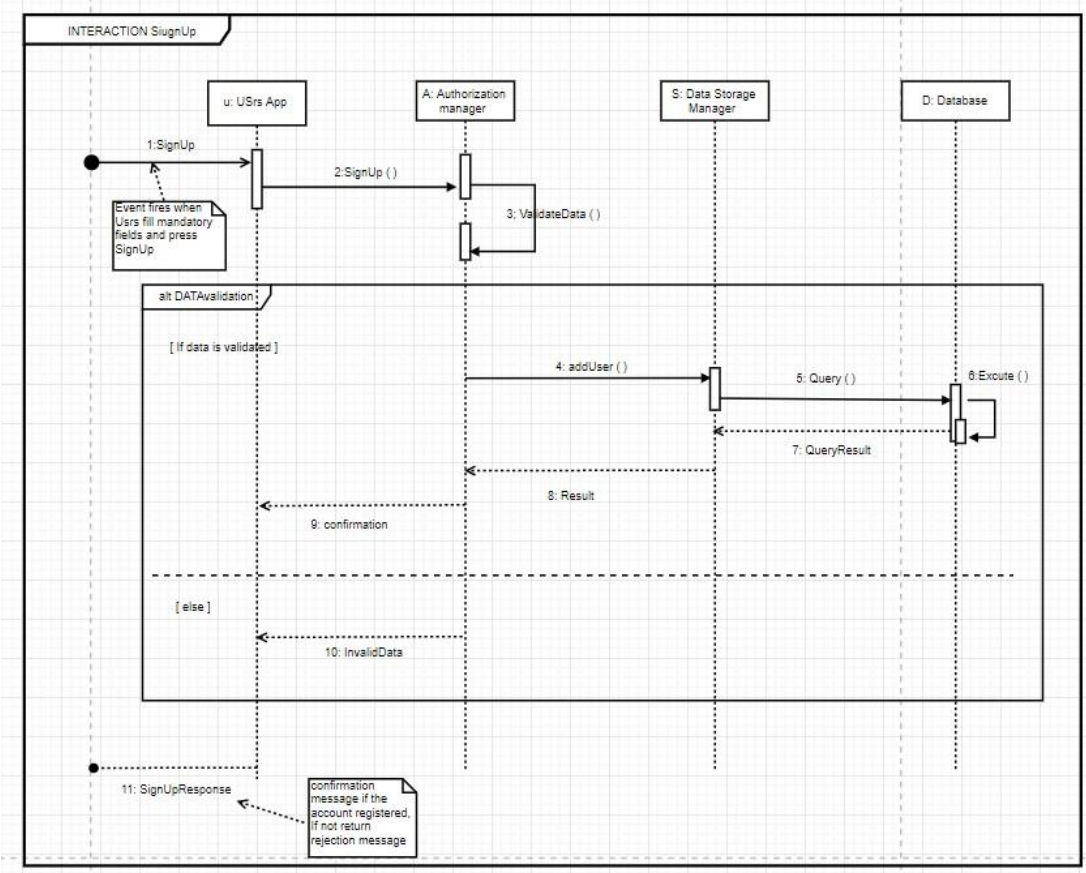


*Figure 3 USrs SignUp Sequence diagram*

### 2.4.2. USrs signs in

Components involved and their roles:

- USrs (Application):

  the activity starts when the USrs, from the website, submits the sign in application, by previously filling out the credentials. When the button “sign in” is pressed, the client calls the method sign in of the Authentication Manager component passing as parameters all the data of the form. He then expects as response either a message with a confirmation of the sign in and the access to the account, or a negative message due to the following situations during the process: invalid data or invalid credentials.

- Authentication Manager

  once the Authentication Manager has been called by the USrs, first check whether the data inserted is valid in terms of type of data, then it calls the Data Storage Manager method getUser through the Data Store Manage component, which is the nexus with the Database.

- Data Store Manager

  It invokes a query to the Database, in order to verify the credentials of the USrs.

- Database

  It verifies if the credentials of the USrs exists and if they are correct.

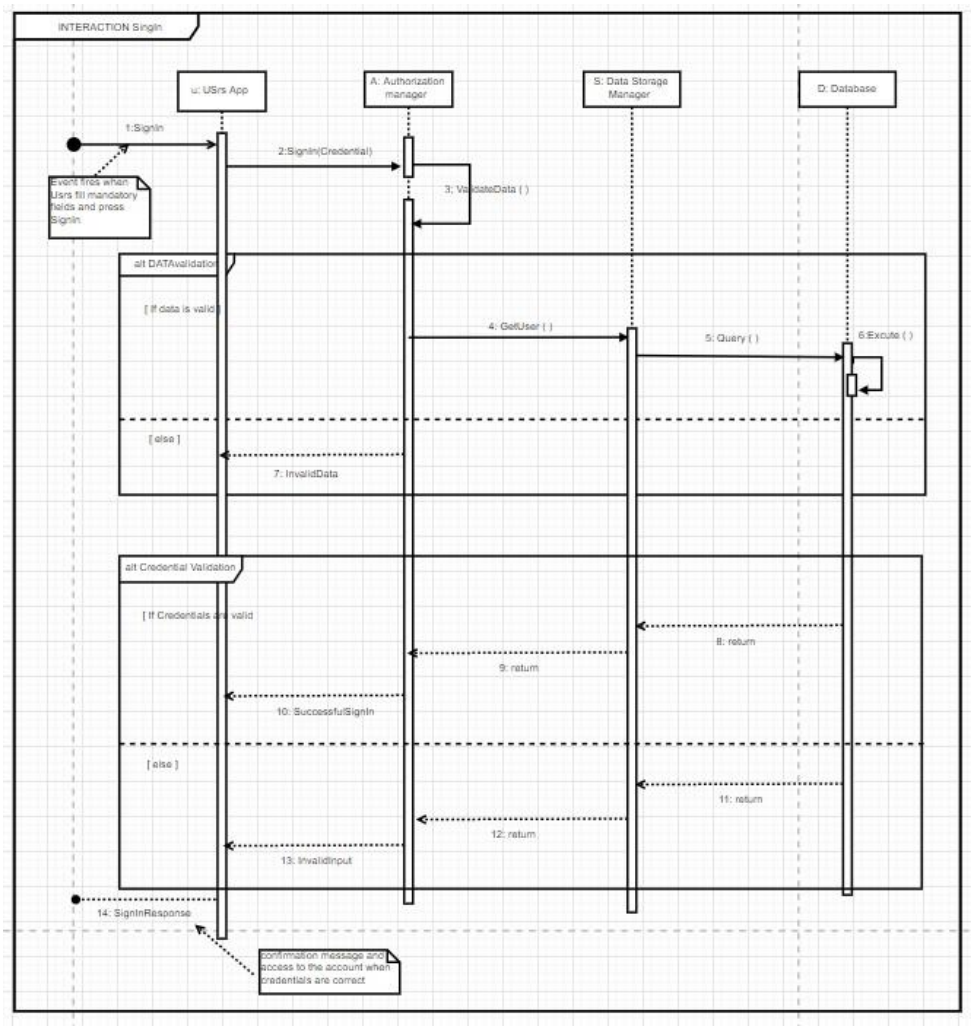


*Figure 4 USrs signIn Sequence diagram*

### 2.4.3. USrs Selects A Service

Components involved and their roles:

- USrs (Application)

  The activity starts when the USrs, from the website, submits the send a request application, by previously filling out all the fields. When the button “create” is pressed, the client calls the method GenerateRequest of the Service Manager component passing as parameters all the data of the form. He then expects as response either a message with the confirmation of the sent SOR or a negative message due to the following situations during the process: invalid data.

- Service Manager

  once the Service Manager has been called by the USrs, then it calls the Data Storage Manager method AddSOR through the Data Store Manage component, which is the nexus with the Database. Additionally, once the storing is confirmed, calls the method SendtoAYLI of the Mailing Manager, in order to send a notification with the information of the SOR

- Data Store Manager

  It invokes a query to the Database, in order to add the SOR in the database.

- Database:

  It stores the information of the SOR.

- Mailing Manager

  it sends the message to the AYLIs with respect to the SOR information.

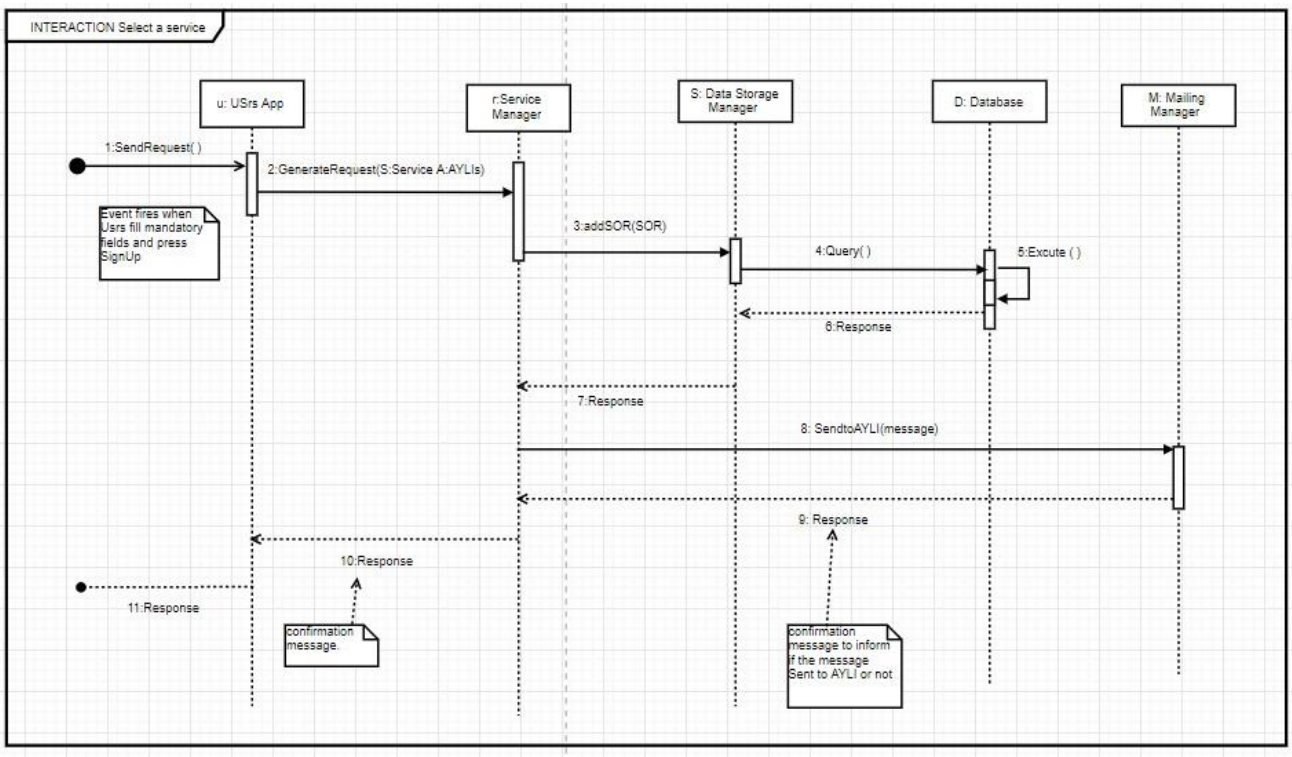


*Figure 5 SendRequest Sequence Diagram*

### 2.4.4. USrs gets the SOR History

Components involved and their roles:

• USrs (Application)

The activity starts when the USrs, from the website, submits the show request history application. When the button “history” is pressed, the client calls the method GetSORList of the Service Manager. He then expects as response a new interface with the list of all the SOR.

• Service Manager

Once the Service Manager has been called by the USrs, then it calls the Data Storage Manager method GetSORList through the Data Store Manage component, which is the link with the Database.

• Data Store Manager

It invokes a query to the Database, in order to get the SOR list of the USrs in the database.

• Database

It returns the list of the SOR related to the USrs.

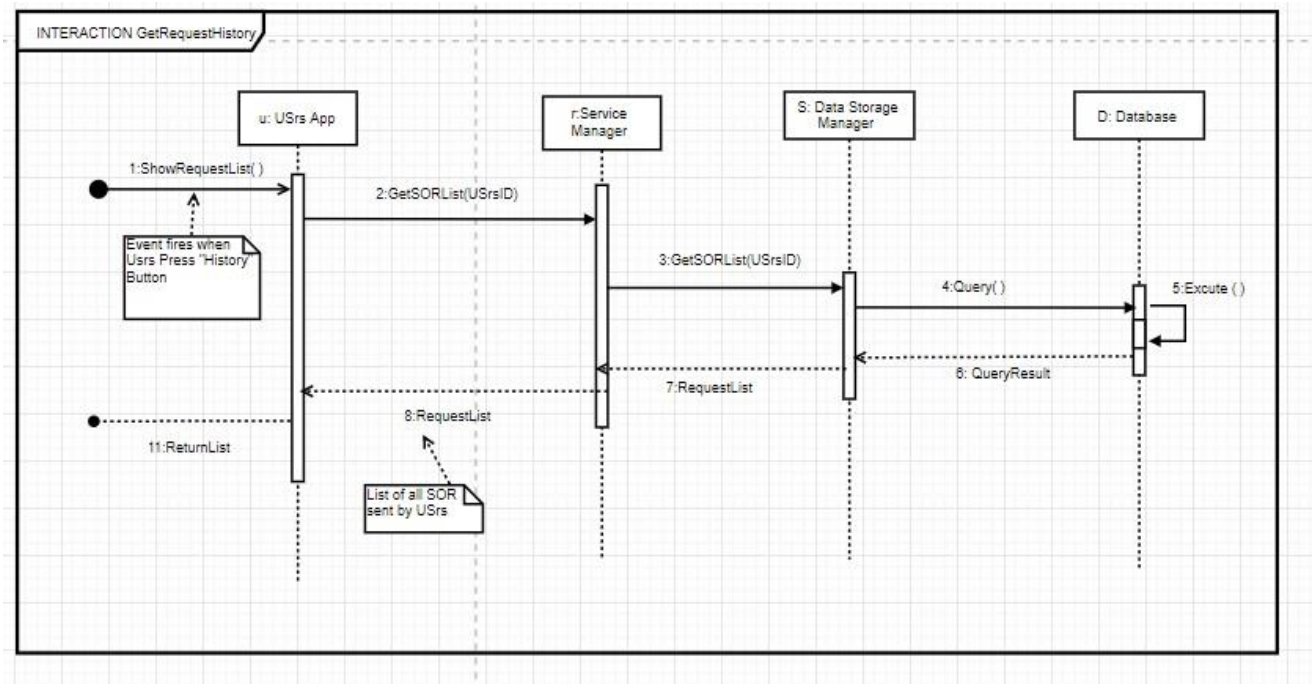


*Figure 6 GetRequestHistory Sequence Diagram*

### 2.4.5. AYLIs sends a Proposal

Components involved and their roles:

• AYLIs (Application)

the activity starts when the AYLIs, from the website, submits the send a proposal application, by previously filling out all the fields. When the button “create” is pressed, the client calls the method GenerateProposal of the Proposal Manager component passing as parameters all the data of the form. He then expects as response either a message with the confirmation of the sent proposal or a negative message due to the following situations during the process: invalid data.

• Proposal Manager

Once the Proposal Manager has been called by the AYLIs, then it calls the Data Storage Manager method AddProposal through the Data Store Manage component, which is the link with the Database. Additionally, once the storing is confirmed, calls the method SendtoUsrs of the Mailing Manager, in order to send a notification with the information of the proposal.

• Data Store Manager

It invokes a query to the Database, in order to update the add the proposal in the database

• Database

It stores the information of the proposal.

• Mailing Manager

It sends the message to the USrs with respect to the proposal information.

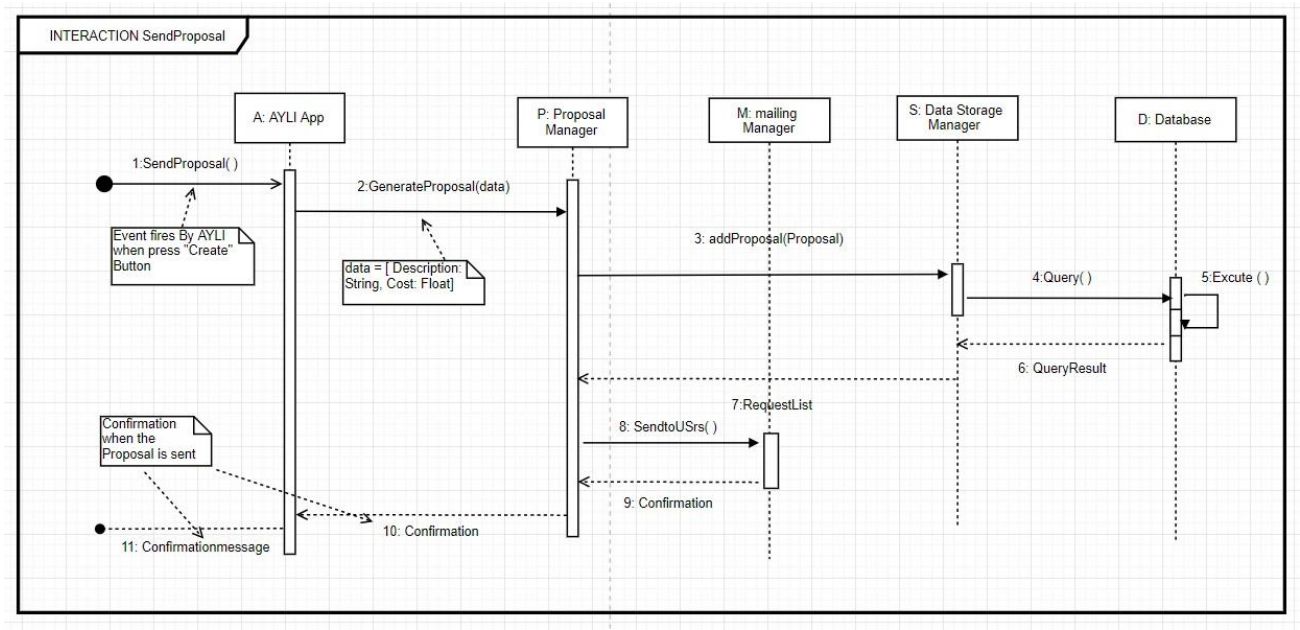


*Figure 7 SendProposal Sequence Diagram*

### 2.4.6. USrs evaluates Proposal

Components involved and their roles:

• AYLIs (Application)

the activity starts when the AYLIs, from the website, submits the send a proposal application, by previously filling out all the fields. When the button “create” is pressed, the client calls the method GenerateProposal of the Proposal Manager component passing as parameters all the data of the form. He then expects as response either an affirmative message with the confirmation of the sent proposal or a negative message due to the following situations during the process: invalid data.

• Proposal Manager

once the Proposal Manager has been called by the AYLIs, then it calls the Data Storage Manager method AddProposal through the Data Store Manage component, which is the link with the Database. Additionally, once the storing is confirmed, calls the method SendtoUSrs of the Mailing Manager, in order to send a notification with the information of the proposal.

• Data Store Manager

It invokes a query to the Database, in order to update the add the proposal in the database.

• Database

It stores the information of the proposal.

• Mailing Manager

It sends the message to the USrs with respect to the proposal information.

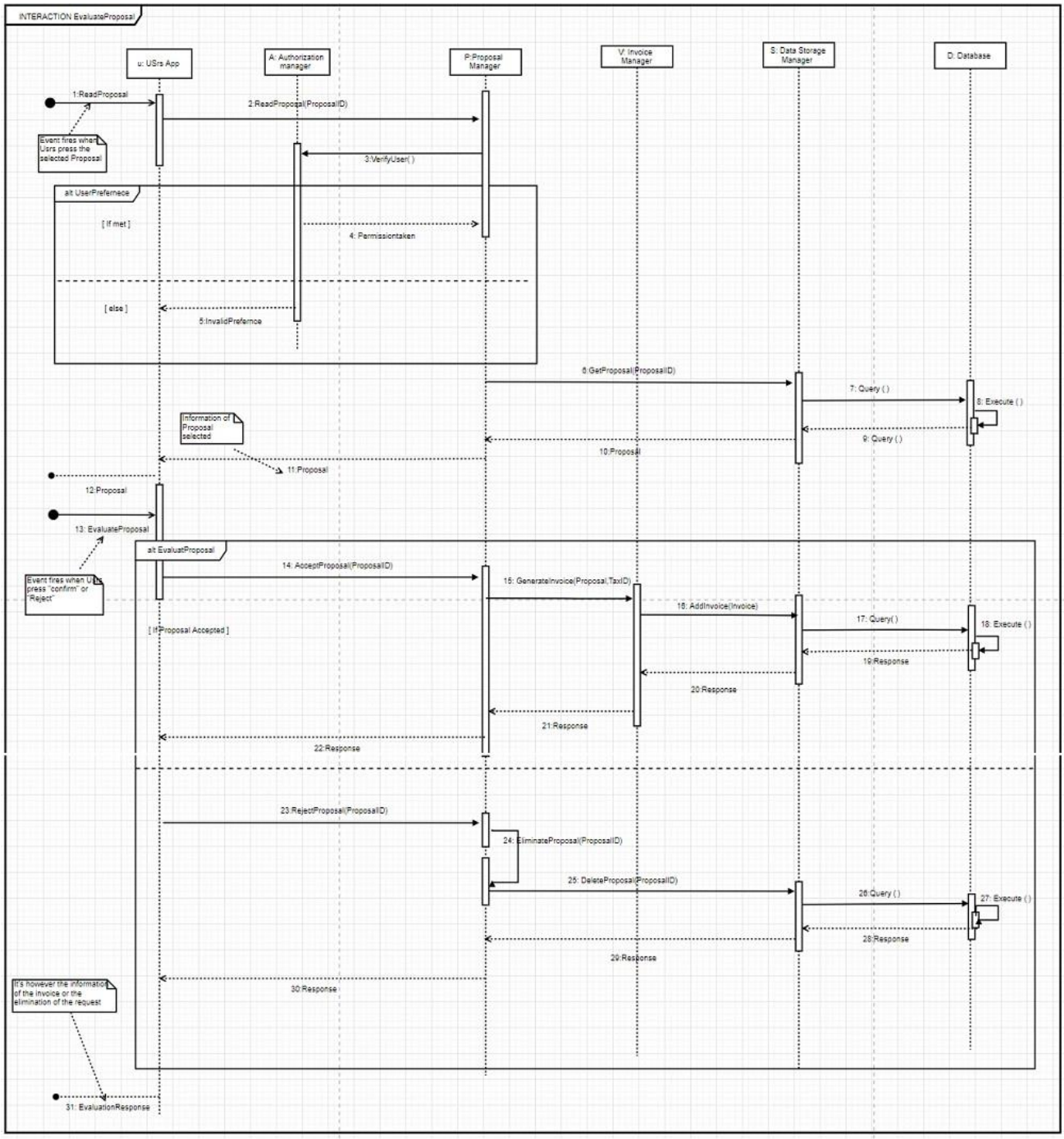


*Figure 8 EvaluateProposal Sequence Diagram*

# 3. User Interface Design

This section explains the user experience either a customer or an AYLI. we will refine the user interface basically from the point of view of interaction with the end-user (mapping between a sequence of actions and a screen flow). As we can see from the following diagrams the leftmost part is devoted to allowing users to sign into the application, and is composed by these elements:

- Sign In
- Sign Up

The main page for the Customer has these choices.

- Send message
- Access to History
- Access to Inbox
- Access to Proposal list
- Request a service from the map
- Access to invoices
- Access to Broadcast
- Log Out

The main Page for the AYLI has these choices.

- Create a new post
- Reply for message
- Create a Proposal list
- Access to Inbox
- Access to shared request
- Access to invoices
- Access to Broadcast
- Log Out

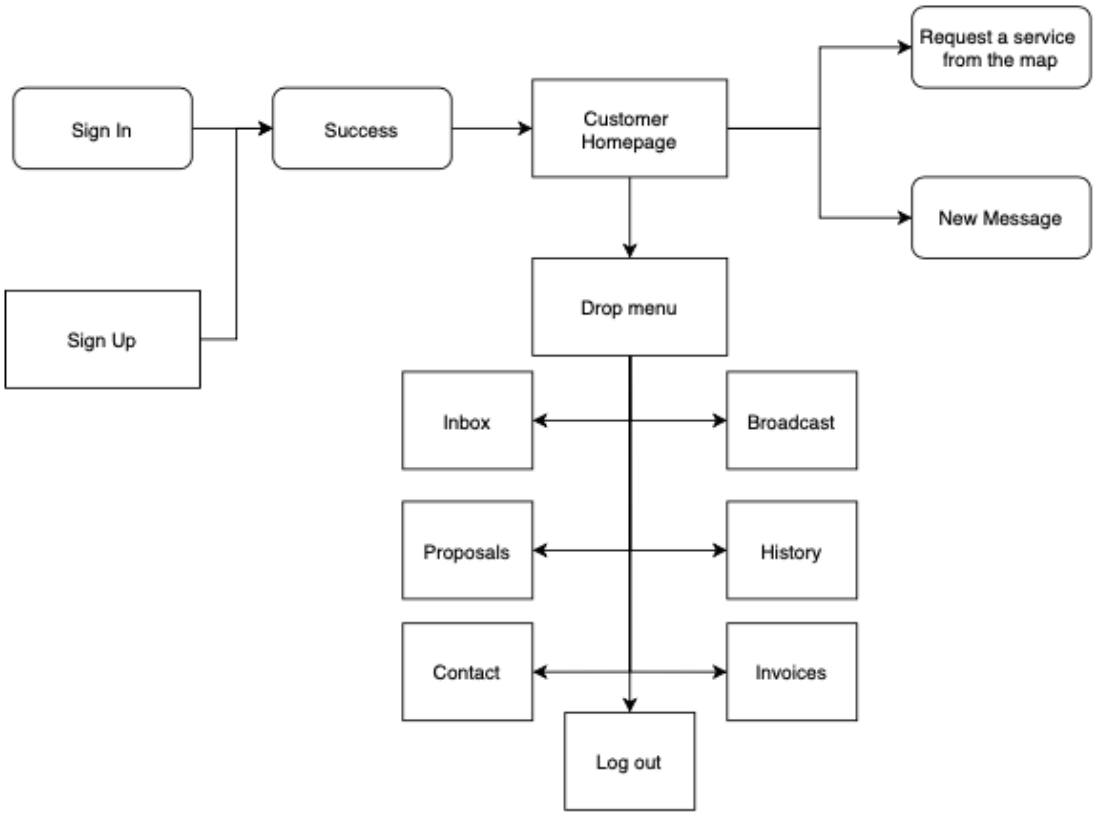


*Figure 9 Customer User Interface diagram*

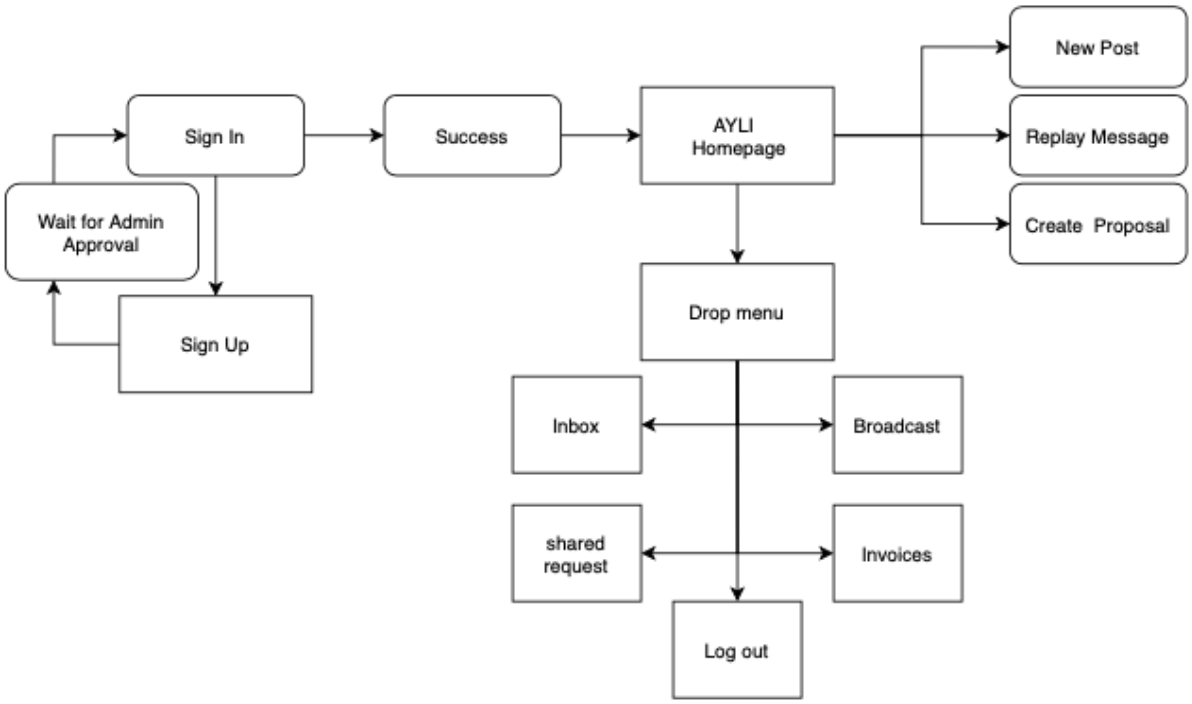


*Figure 10 AYLI User Interface diagram*