I3301,I3302,I3350

Supervisors:  
Dr. Charbel Nicolas  
Dr. Ziad EL Balaa  
Dr. Bernadette Wakim

Presented By:

Carlita Khawand, 55028  
Yorgo Wakim, 54689

[1- **Introduction** **Error! Bookmark not defined.**](file:///D:\Studies\Computer%20Science\3rd%20Year\Projects\I3301,%20I3302,%20I3350\Plan%20du%20Projet.docx#_Toc465704774)

[2- **Analysis** 4](file:///D:\Studies\Computer%20Science\3rd%20Year\Projects\I3301,%20I3302,%20I3350\Plan%20du%20Projet.docx#_Toc465704775)

[2.1- Business Rules 4](file:///D:\Studies\Computer%20Science\3rd%20Year\Projects\I3301,%20I3302,%20I3350\Plan%20du%20Projet.docx#_Toc465704776)

[2.2- Use case 5](file:///D:\Studies\Computer%20Science\3rd%20Year\Projects\I3301,%20I3302,%20I3350\Plan%20du%20Projet.docx#_Toc465704777)

[2.3- Class Diagram 9](file:///D:\Studies\Computer%20Science\3rd%20Year\Projects\I3301,%20I3302,%20I3350\Plan%20du%20Projet.docx#_Toc465704778)

[2.4- Activity Diagram 11](file:///D:\Studies\Computer%20Science\3rd%20Year\Projects\I3301,%20I3302,%20I3350\Plan%20du%20Projet.docx#_Toc465704779)

[2.5- Sequence Diagram 12](file:///D:\Studies\Computer%20Science\3rd%20Year\Projects\I3301,%20I3302,%20I3350\Plan%20du%20Projet.docx#_Toc465704780)

[3- **Implementation** 15](file:///D:\Studies\Computer%20Science\3rd%20Year\Projects\I3301,%20I3302,%20I3350\Plan%20du%20Projet.docx#_Toc465704781)

[3.1- Technical Encvronment 15](file:///D:\Studies\Computer%20Science\3rd%20Year\Projects\I3301,%20I3302,%20I3350\Plan%20du%20Projet.docx#_Toc465704782)

[3.2- Screens Description 16](file:///D:\Studies\Computer%20Science\3rd%20Year\Projects\I3301,%20I3302,%20I3350\Plan%20du%20Projet.docx#_Toc465704784)

[4- **Conclusion** 23](file:///D:\Studies\Computer%20Science\3rd%20Year\Projects\I3301,%20I3302,%20I3350\Plan%20du%20Projet.docx#_Toc465704785)

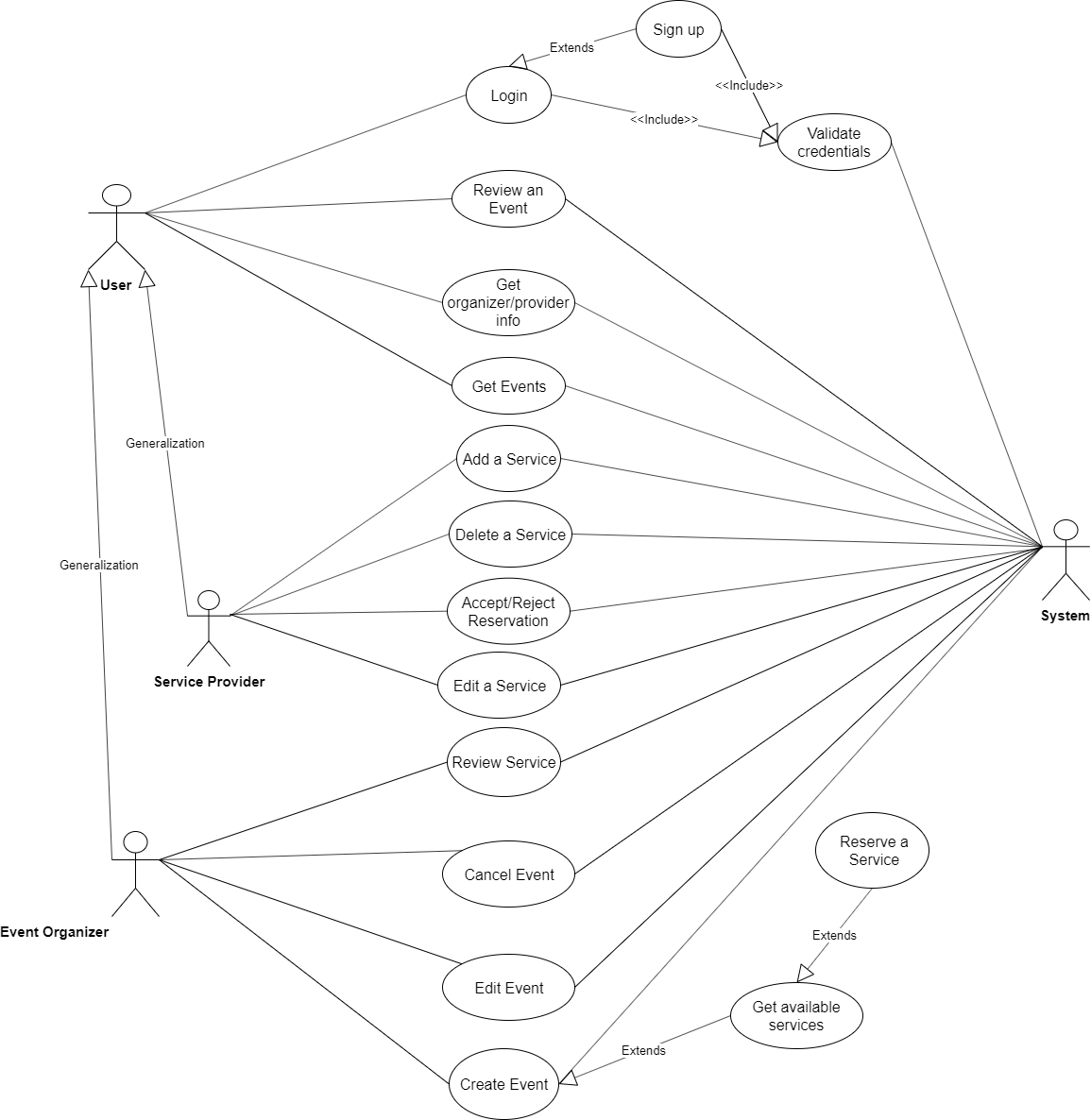
***1-Introduction:***  
The most trending business in Lebanon and a huge number of other countries is the events business. One creates an event, sells the tickets and makes a decent amount of money. With the diversity of the services needed to organize an event, it becomes difficult to have an overview of everything available to pick the best options.  
  
With Save The Date, you can organize your own event, and provide it with any feature you want, without missing any of the appealing deals and offers you can find, all grouped in a single and user friendly application and website.  
  
Let’s take a quick look at the details.

***2-Analysis:***

**2.1-Business Rules:**

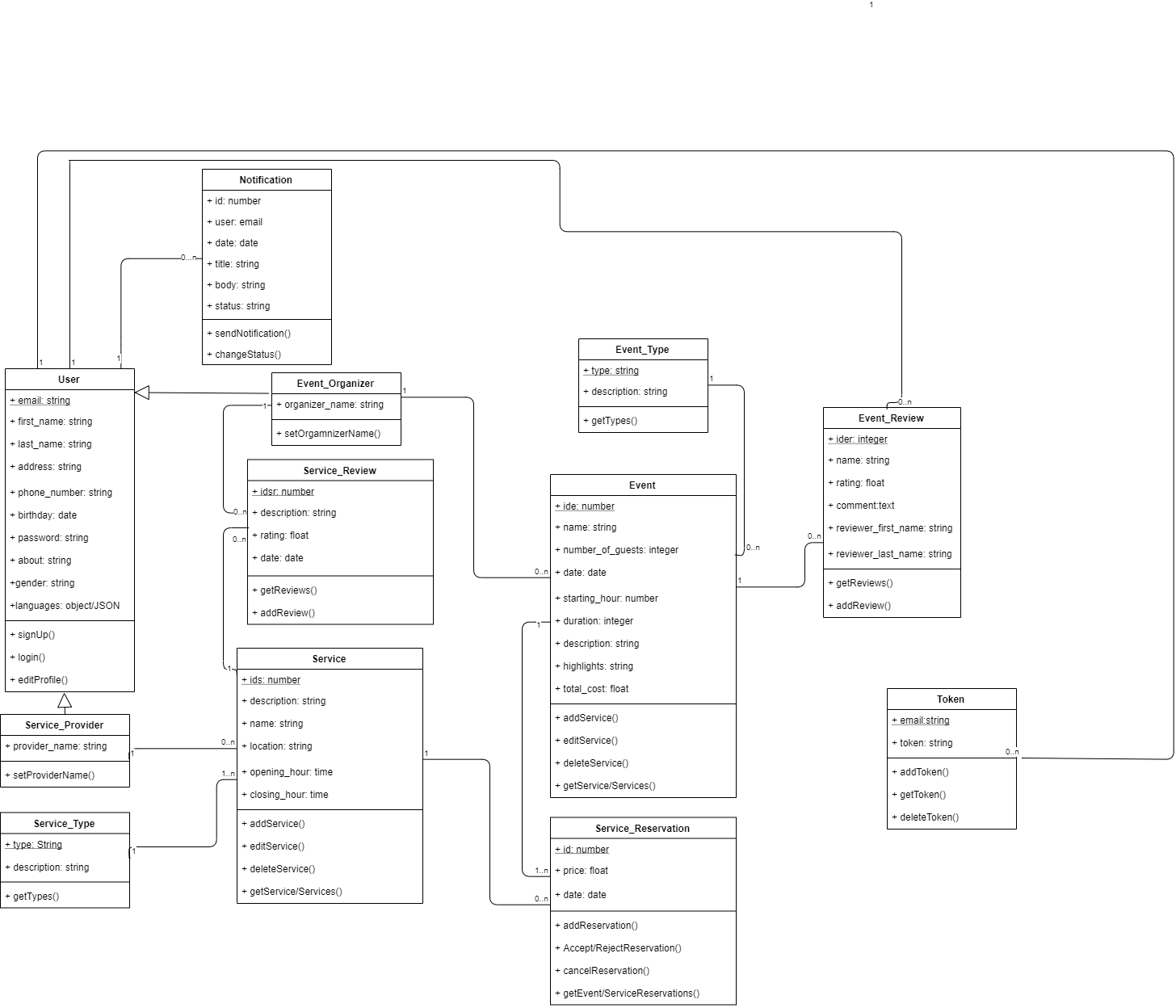
* The user can see all the events and their organizers, their providers, and see all the info related to them.
* For anything else, the user needs to log in.
* While organizing an event, the event organizer should contact the provider of any desired service, then the organizer initiates a reservation. At this moment, the service provider can either reject the reservation, or set the price agreed on and accept it.
* The cost of any event is the sum of the costs of all its accepted services reservation. If any reservation is cancelled, its price is withdrawn from the total cost of the event.

***2.2-Use Case Diagram:***

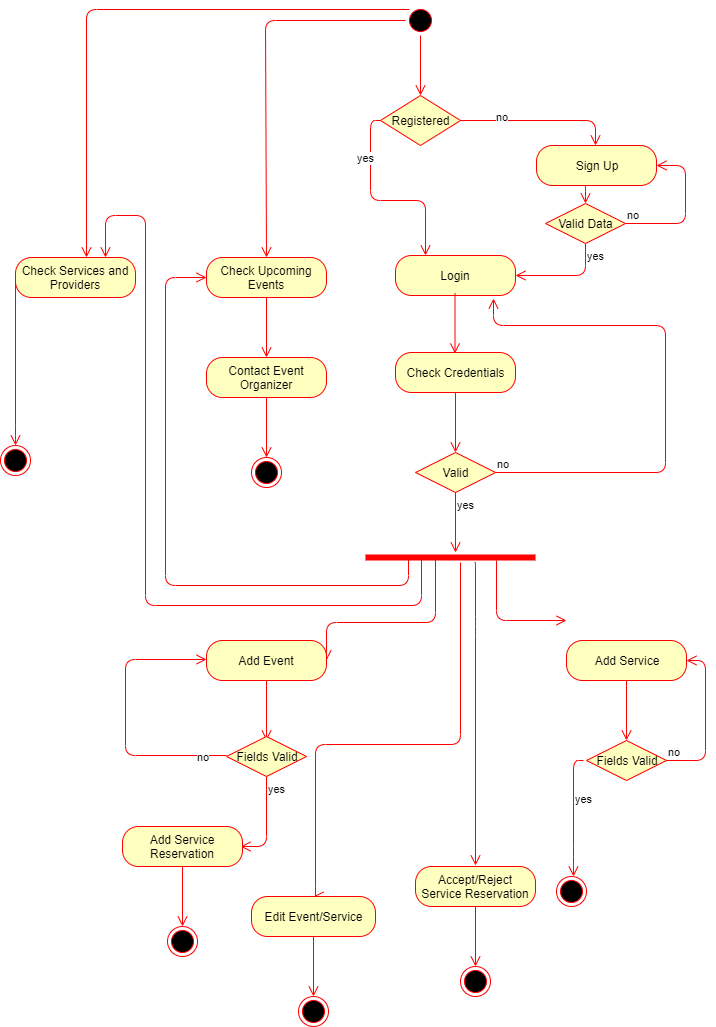
****

* Login:  
  Condition: The user should have an account.  
  Description: The user inputs his email and password, and the system logs him in if the credentials are valid.
* Sign Up:  
  Condition: None.  
  Description: The user fills in the sign up form, and if the data is valid,  
  an account is created for him, and he is automatically logged in.
* Get Events:  
  Condition: None.  
  Description: The user can see the available upcoming events.
* Create Event:  
  Condition: User must be logged in.  
  Description: User enters the event name and details, and once he submits,  
  the event will be saved and displayed.
* Edit Event  
  Condition: The event exists and the user is its organizer.  
  Description: User changes any field he wants to, even the picture, and saves it into the database.
* Cancel Event:  
  Condition: The event exists and its date hasn’t passed yet.  
  Description: The user cancels the events: it is deleted and will not be displayed anymore.
* Reserve a service  
  Condition: The date of the selected event hasn’t passed yet.  
  Description: When creating or editing an event, the user is able to add a service reservation (catering, DJ…) to its event, after agreeing on the price with the event organizer.
* Cancel Reservation  
  Condition: The event hasn’t taken place yet.  
  Description: Organizer removes this service from its event.
* Accept/Reject Reservation  
  Condition: User must be the provider of the reserved service.  
  Description: The provider either rejects the reservation, or sets the price agreed on and accepts it.
* Get Available Services  
  Condition: None.  
  Description: User can see the available services.
* Review Event  
  Condition: Must be logged in.  
  Description: The user can add a rating and a review to an event.
* Review Service:  
  Condition: Must be an event organizer.  
  Description: The user can add a rating and a review to a service.
* Get Organizer/Provider Info  
  Condition: None.  
  Description: See the profile of the event organizer.
* Add Service  
  Condition: Must be logged in.  
  Description: User can provide a service for the event organizer to benefit from. He fills out the form fields, and the service is published once it’s saved.
* Edit Service  
  Condition: The service exists and the user is its provider.  
  Description: User changes any field he wants to, even the picture, and saves it into the database.
* Delete Service  
  Condition: The service exists and the user is its provider.  
  Description: The user deletes the services: it is deleted and will not be displayed anymore.

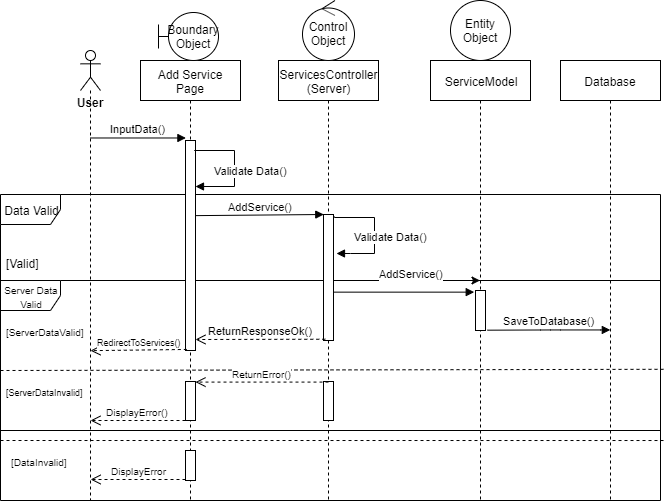
***2.3- Class Diagram:***

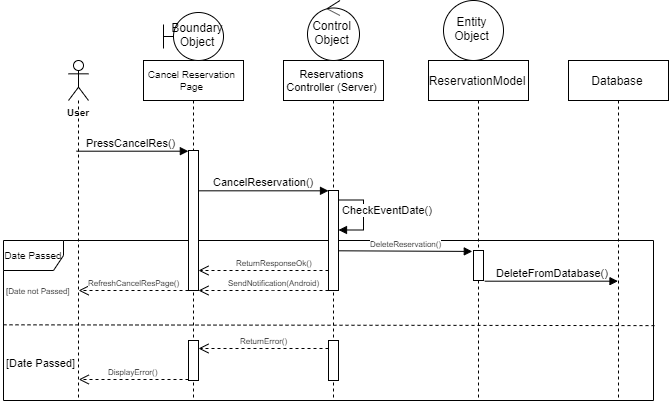
******

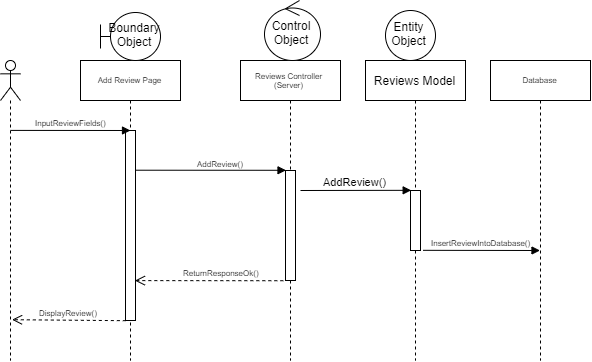
**Explications:**-A user can have many tokens (In case of multiple device login), but a token only belongs to one user.  
- A user can write many reviews, but a review can be written by only one user.  
- A user can receive many notifications, but a notification is only sent to one user.  
- A user can be a normal user, an event organizer, a service provider all of them.  
- An event organizer can organize many events, but an event can only be organized by one event.  
  
- The organizer can write many reviews on a service, but a service is written by only one organizer.  
- An event has only one type, but there can be many events of the same type.  
- The event review belongs to only one event, but an event can be reviewed many times.  
- An event can make many service reservations, but a reservation can only belong to one event.  
- A reservation can only be made on one service, but a service can be reserved many times.  
- A provider can provide many services, but a service is provided by only one provider.  
- A review can be made on only one service, but a service can be reviewed many times.  
- A service has only one type, but there can be many services of the same type.

***2.4- Activity Diagram:***

***2.5- Sequence Diagrams:***

***  
Title: Add service Sequence Diagram***

***  
Title: Cancel Reservation Sequence Diagram***

***  
Title: Event Review Sequence Diagram***

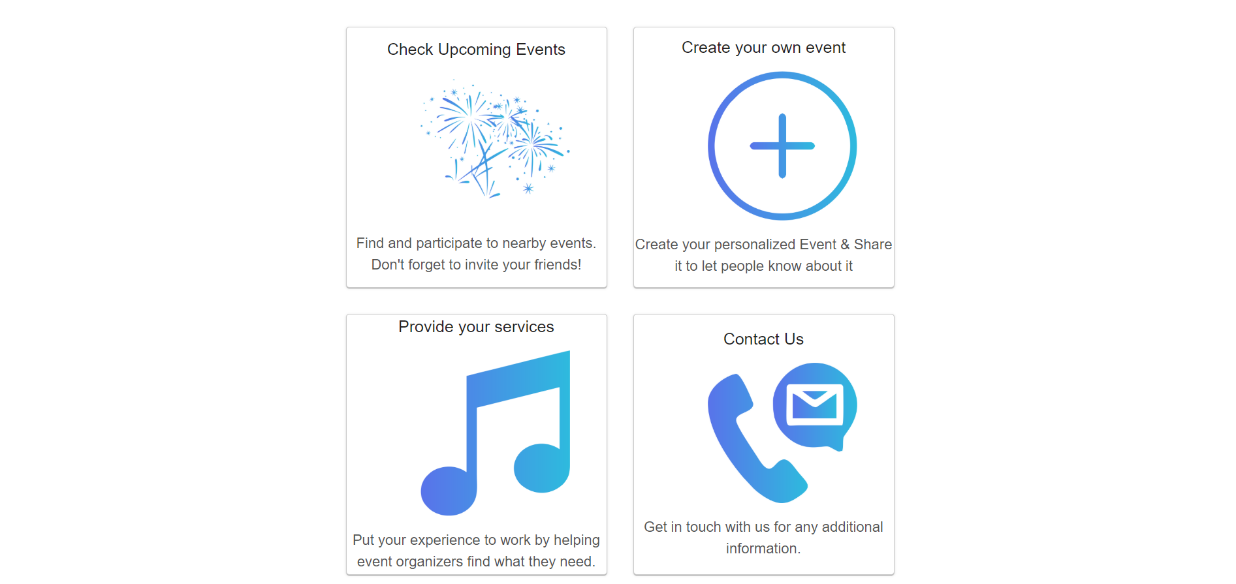
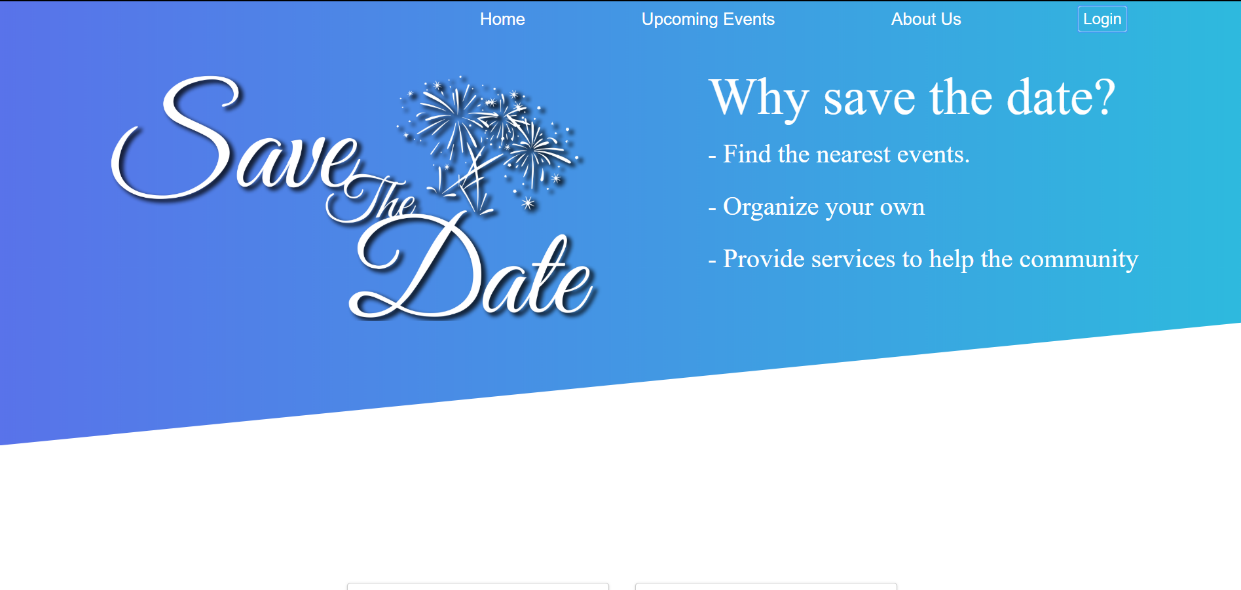
***3-Implementation:***

***3.1-Environment Used:***This project is built on a raw PHP backend (because no frameworks were allowed), which is an API modeled following the MVC architecture, emphasizing that the Views are the Android mobile application, and the Angular 9 single page application for the website.  
The development was made using the Visual Studio Code editor for both the backend and the web view, and Android Studio for the mobile application.

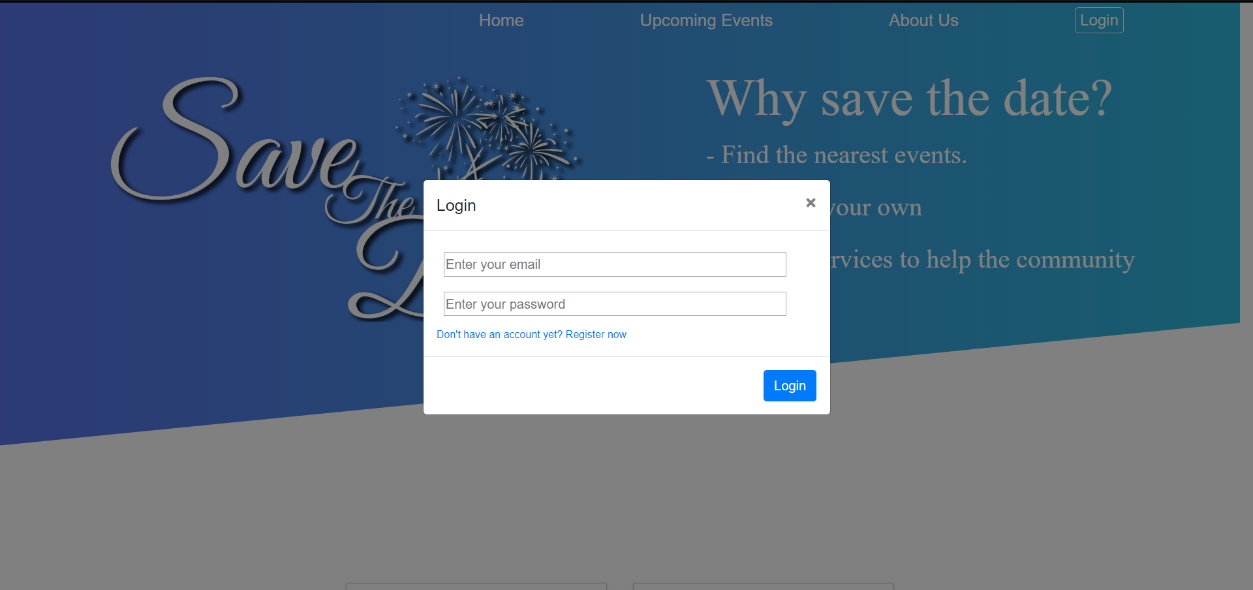
Details of our MVC Architecture:  
In our backend we have Controllers that receive the request, process it using the models and utilities we created, and send back JSON response data to both our views (Android and Angular).

**NB: We used Draw.io to create all the UML Diagrams.**

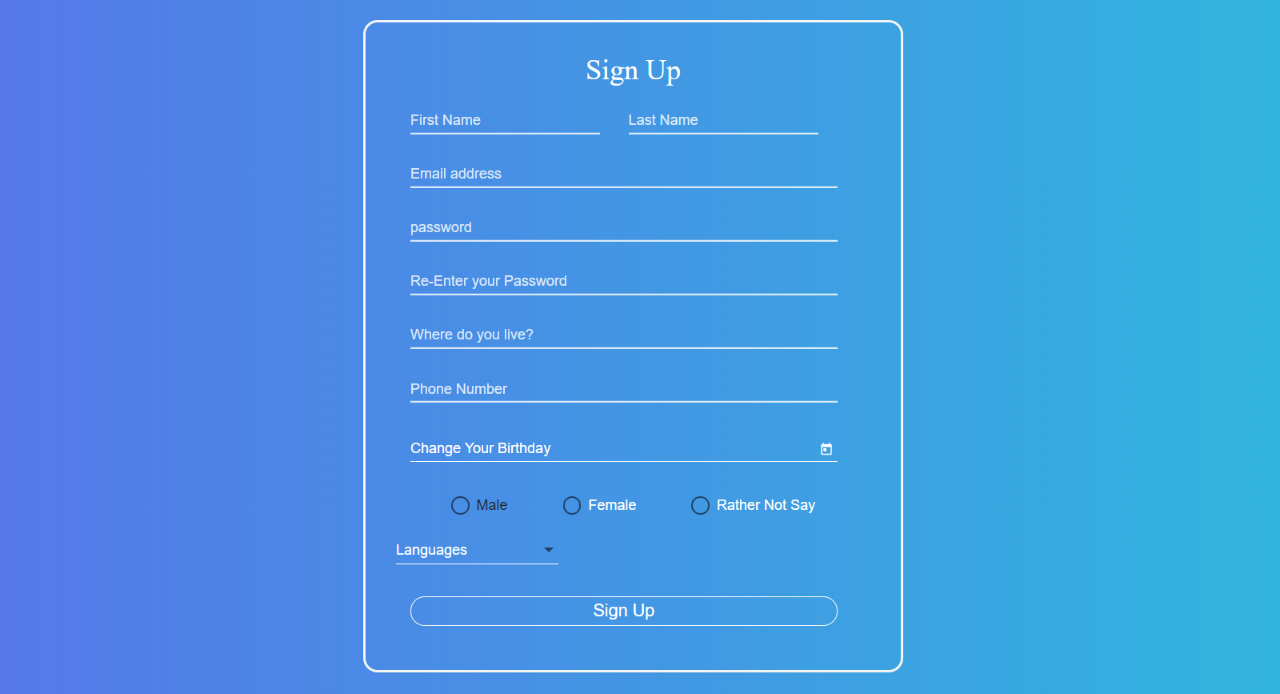
***3.2-Screens Description:***



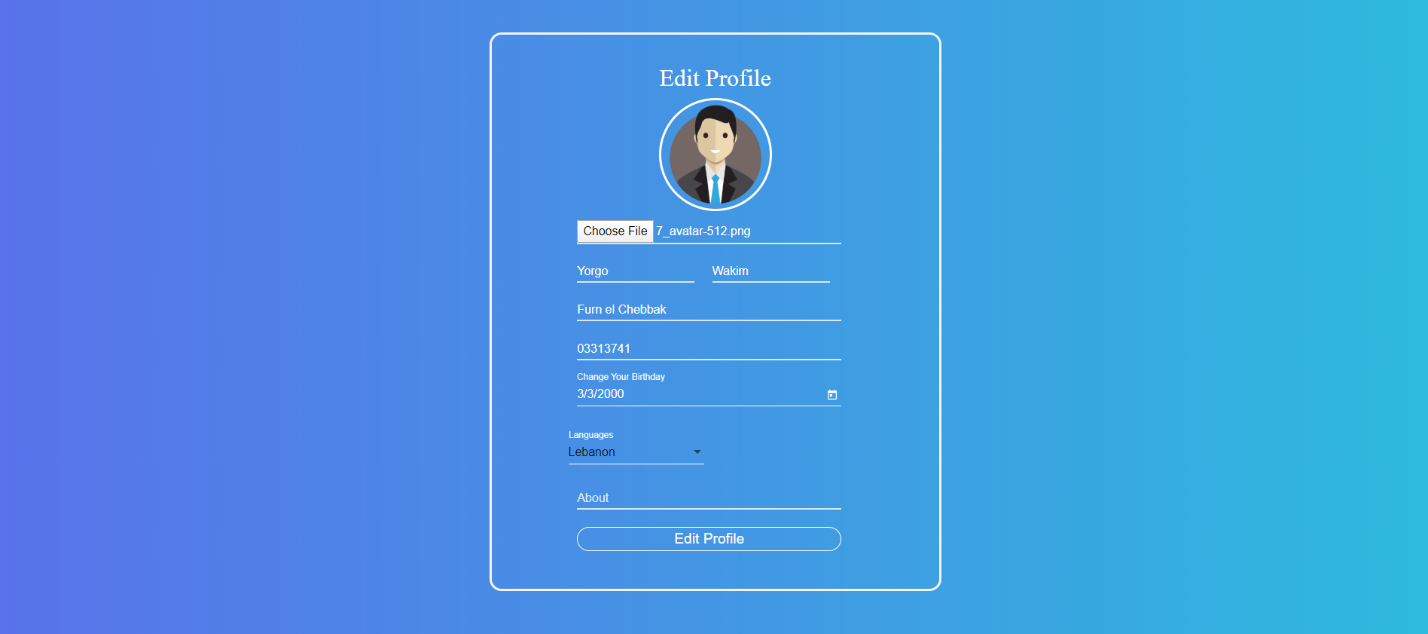
**Landing Page:**  
This is the page the user lands on upon visiting our website. It uses the User Controller, Event and Service controller. The user will be redirected to any page upon request.

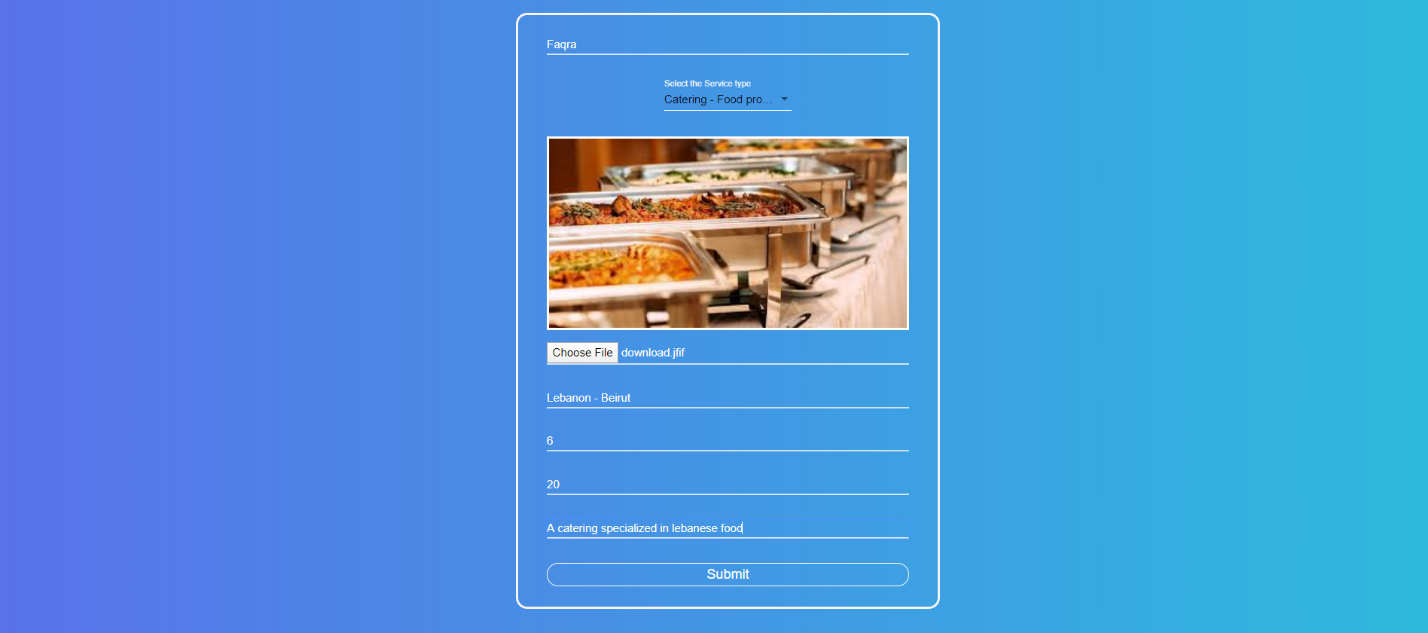


**Login Modal:**  
The user inputs his email and password, and the credentials are validated in the user controller, to receive an authentication token (Similar to JWT token, but custom made since we can’t use any framework), and the token will be saved in the local storage, then he will be redirected to the upcoming events page. He can also navigate sign up in case he doesn’t have an account.

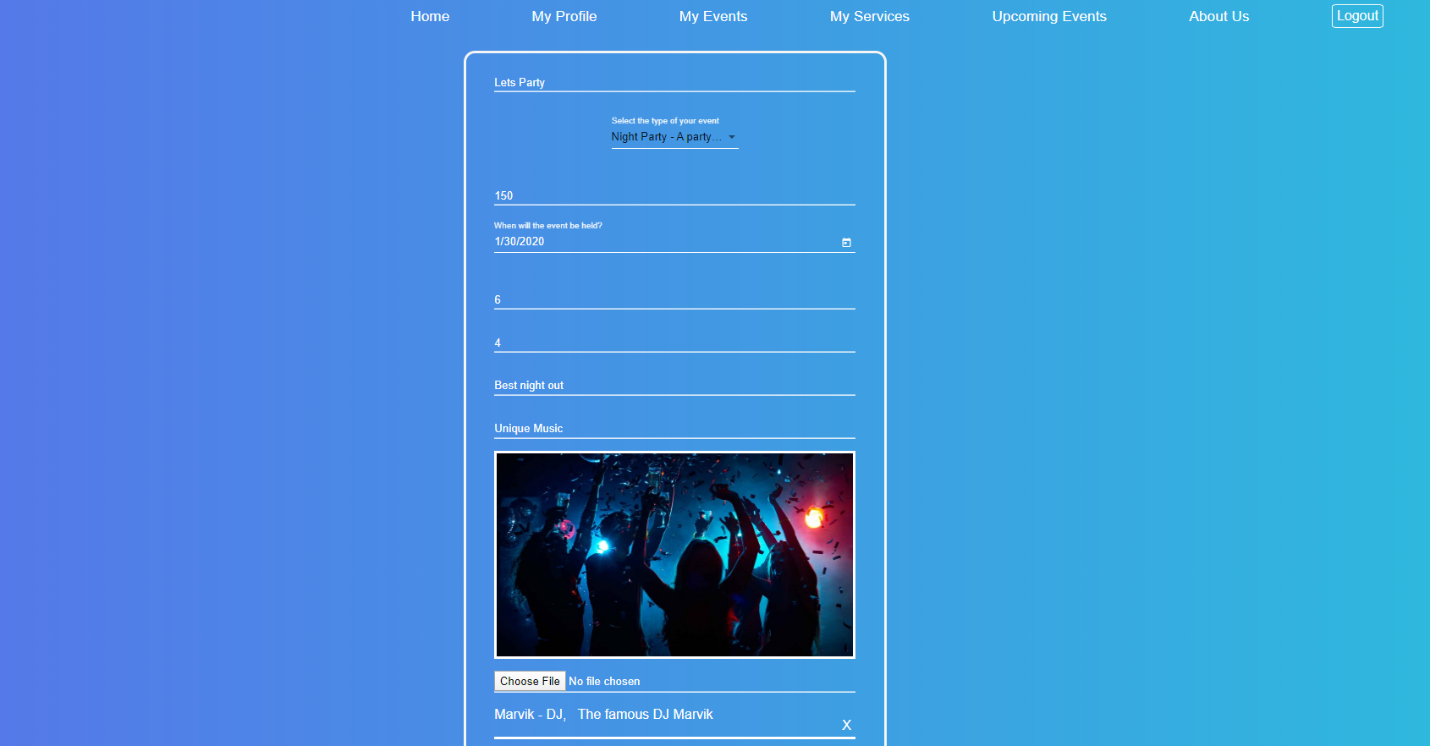


**Sign Up Page:**  
The user has to fill in all the fields then when pressing sign up, his data will be saved to the database using the User Controller, log him in automatically and he will be redirected to edit his profile.

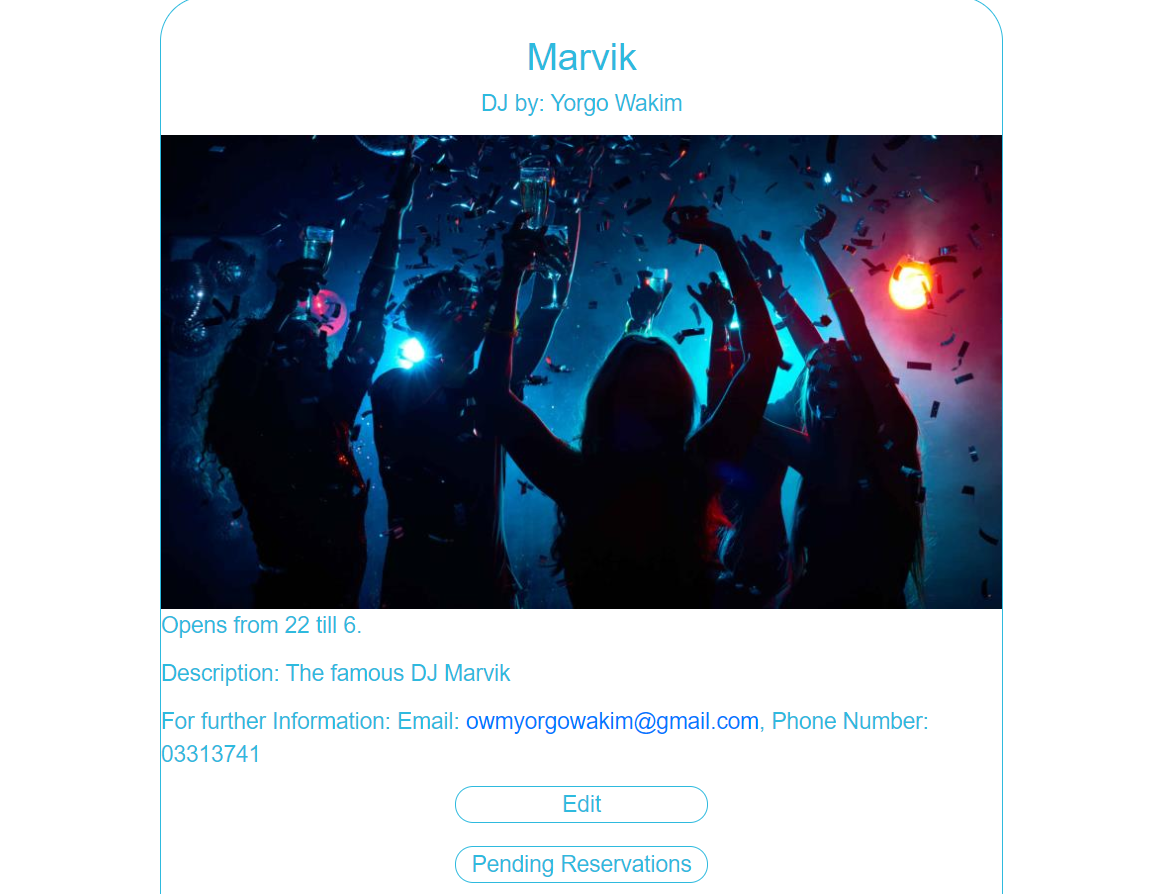
  
**Edit Profile Page:**  
The user can edit previously entered information, and he also has to set a profile picture, and a little description about him. Then the data will be validated in the User Controller, and he will be redirected to the upcoming events page.



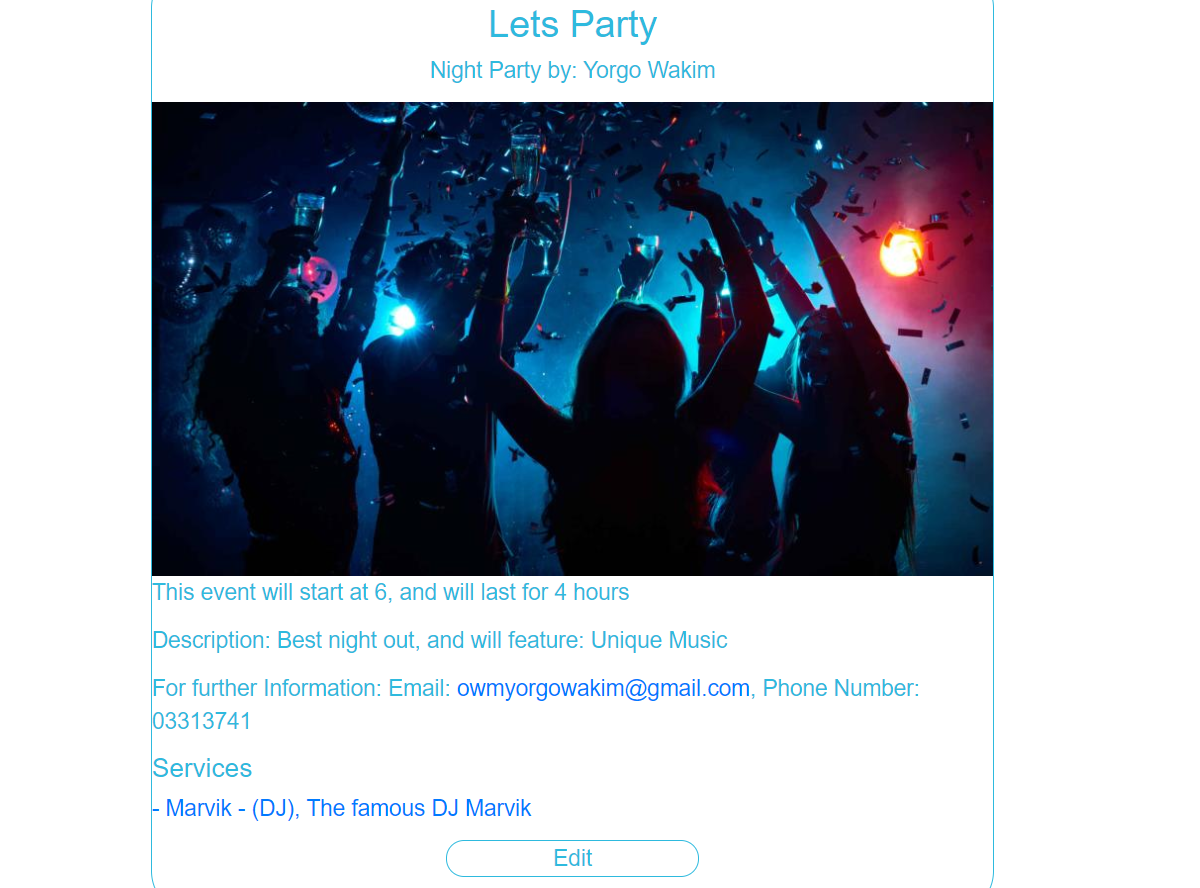
**Add/Edit service Page:**  
The user can add a service that will be visible to everyone. Once he inputs all the required data, it will be saved to the database using the Service Controller. In case of editing, the same procedure will happen, but the old data will fill the form upon loading the page.

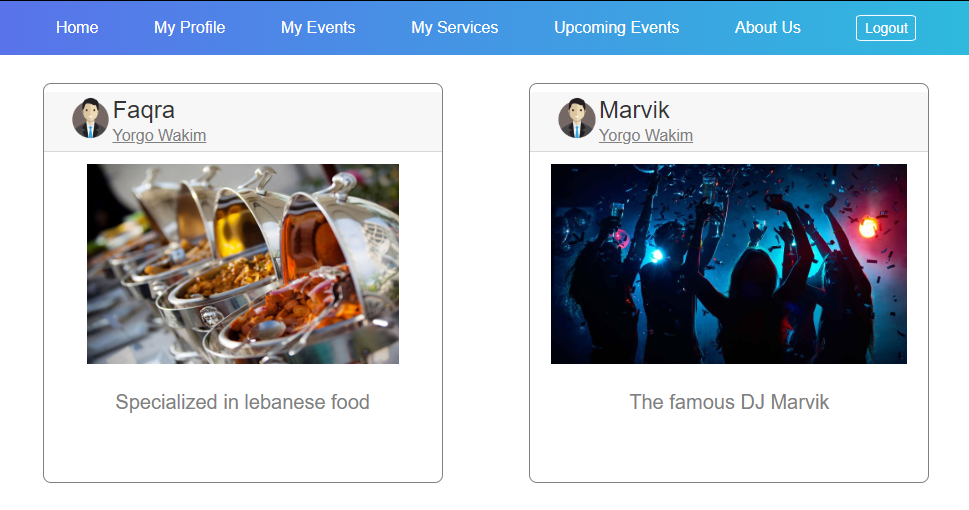


**Add/Edit Event page:**  
The user can add an event that will be visible to everyone. Once he inputs all the required data, it will be saved to the database using the Event Controller. In case of editing, the same procedure will happen, but the old data will fill the form upon loading the page. In addition to that, he can add service reservations to his event, or delete the ones he made (In case of editing), using the Reservation Controller.

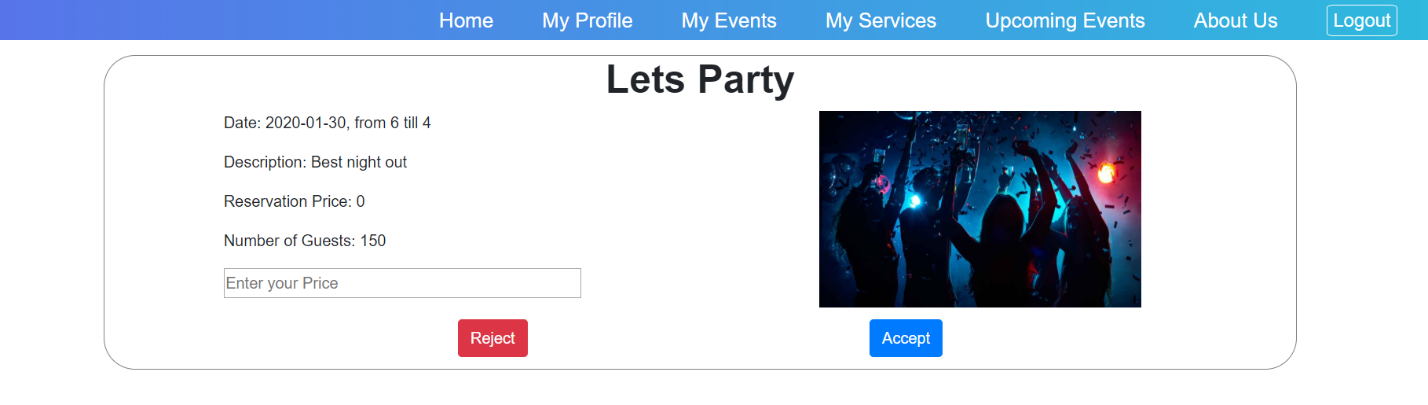


**Single Service Page:**  
User can see the details of a service, which is fetched by the Service Controller. If the user that accesses this page is the provider of the service, he will be able to edit it, or see its pending reservations.

 **Single Event Page:**  
User can see the details of n event, which is fetched by the Event Controller. If the user that accesses this page is the organizer of the event, he will be able to edit it.



**Events/Services List:**  
This is how the events/services list will be displayed, fetched by the Event/Service Controller.



**Events Reservations Page:**  
The service provider will be able to see the event that reserved the service, he can reject the reservation, or input the price agreed on and accept it.

***4-Conclusion:***The bottom line says that the difficulties faced to organize an event are solved with this unique system, since the whole world is trending with attending events, even coming up with new types of events never seen before.

Furthermore, there is no learning new things without struggles. We faced little problems that we solved successfully in a number of technical issues:

1. Deciding how to send the image across the network using http requests, and how to save it in the database, as a base 64 string, or save it on the server and save its path in the database.
2. Coming up with a way to implement the JWT logic in a simpler way for authentication, since we were not allowed to use any framework, and since the authentication in angular should be done using a token. (We could have used the session id as token but it would have been a bit of time waste to create and manage sessions).
3. Sending a successful http request from the android app and listening to the response in the right way.