BABEŞ-BOLYAI UNIVERSITY CLUJ-NAPOCA

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

SPECIALIZATION COMPUTER SCIENCE - ENGLISH

DIPLOMA THESIS

WEDDING BOOKING

Scientific Coordinators:

University Assistant
Dr. Bădărînză Ioan

Author:

Soporan Darian-Dorel

Cluj-Napoca

2020

UNIVERSITATEA BABEȘ-BOLYAI CLUJ-NAPOCA

FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

SPECIALIZAREA INFORMATICĂ – ENGLEZĂ

LUCRARE DE LICENŢĂ

WEDDING BOOKING

Conducători ştiinţifici :

Asistent Universitar
Dr. Bădărînză Ioan

Absolvent:

Soporan Darian-Dorel

Cluj-Napoca

2020

Abstract

Wedding Booking is the most comprehensive wedding site in Romania and also the first one. This offers the ability to find the best vendors for your budget and to plan your wedding exactly the way you envisioned it. We make it simple for you to get the pricing, the availability and get in contact with vendors across every town. Inspired by **Airbnb** and **Booking.com ,** in addition to the wedding industry's best and brightest wedding vendors, Wedding Booking come up with a suggestion algorithm, guided by the **Facebook** and **Instagram** engaging algorithm, that propose the best services of a vendor, regarding its engagement, quality and nature.

The first chapter in our journey is about the difficult part of searching and finding the perfect vendors for the perfect wedding day. At start, I will describe and present what Wedding Booking is and why is it likely to be used nowadays by Wedding Service Providers and Married to be's. Then I will state some of the advantages of using our website.

Chapter two illustrates the architecture of our application, such as: what we have used for persisting data and how me managed to link the frontend to backend. I will present the programming languages that were used in both application parts stated above along with the technologies. Furthermore, I will show some of the design patterns that I integrated in my application along with the algorithm for service and package suggestion.

In chapter three there will be a user manual where it will be presented the instructions on how a user should control and use our application. I will expose the stages of how a client can book one or more services and how an income producer can state and post their multiple services. Along with the database diagrams, we will detail our application functionalities using use case and sequential diagrams.

Meanwhile, our team of vendors will provide users the best wedding ideas filled with inspiration to help you choose between location , DJ's, photographers and more when you start planning the details.

The engagement suggestion algorithm along with the application was proposed at "Comunicări Științifice ale Studenților – Informatică 2020". This work is the result of my own activity. I have neither given nor received unauthorized assistance on this work.

TABLE OF CONTENTS

1.Introduction

One of the foremost significant days in a person's life is the wedding day! To ensure that such a momentous event goes off without a hitch, appropriate planning is required. Months are going to be spent prior to the marriage researching vendors, comparing products and costs, meeting with photographers, hotel managers, and lots of wedding vendors to determine how to save costs on each wedding item. Without an implementation plan that outlines budgets, how important tasks should be organized, and what the contingencies are within the event that something goes awry, could end up in what should be one of the happiest days of someone's life turning into a complete disaster. The success of your whole day depends on your ability to prepare, organize, plan, and budget.

The stakeholders in a wedding project are numerous. Naturally the bride and the groom are the largest stakeholders. But also included during this list of stakeholders are going to be the bride and groom's family, members of the marriage party (maid of honor, bridesmaid, best man, and groomsmen), wedding guests, wedding vendors (florist, musicians, caterer, cake maker, clothiers, venue directors, dry ice machines, magic mirror), and therefore officiate. Further, a successful wedding requires the orchestration of the many moving parts and therefore the availability of the many resources including people, skills, equipment, materials, and working capital. (1)

A wedding budget is completely essential to planning a marriage and in fact should be one of the very first things that a marrying couple should think about. Further, a typical wedding takes months to plan. There are arrangements to make, reservations to secure, items to order, and logistics to figure out.

Before getting too far into the wedding plan, one will want to determine a date for the big event. Most of the major decisions that will need to be made from this point are going to be contingent on the wedding date. Selecting the proper wedding date also requires more than picking a date at random. Most couples prefer the spring and summer months over the autumn and winter months because of to the warmer weather. Some couples even have a holiday preference and look forward to getting married on Valentine’s Day or perhaps Halloween.

When determining time scope, it might be helpful to begin with the last activity, during this case the wedding day itself, and work backwards towards the beginning of the project. The logical sequence of tasks are going to be organized by determining what must be completed immediately before each successive task. Once the project network has been laid out, one should check forward by confirming that every activity is the only task necessary immediately before starting the following activity. (2)

Wedding Booking – Short Description

Wedding Booking is a full service platform which offers complete services for weddings . Here our married to be can search through a range of posts from small details, as dry ice for the dancing part, all the way to booking the perfect wedding location. The posts are uploaded by people who can provide those types of services and want to promote their personal company. We are conscious of people needs and work with them to come up and organize the event of their dreams. Our goal is to put the fun back into planning a marriage. Too many of us become overly stressed and frustrated when planning these wonderful events. Wedding is a dream, wedding is an experience and wedding is also a trouble for newlyweds. Fortunately, with our platform, new couples are being able to enjoy the whole process without troubles.

Objectives

Whether this is our customer’s first wedding or not, we would like that each detail of their event to be both a pleasant and memorable experience. With our platform the customers doesn’t have to stress about anything, we will provide everything they need.

Role

Our mission is to maintain our customers satisfied! Weddings can be very traumatic and time consuming. We’re here to take the heaviness of the customer so that they could spend extra time off with the family. We hear their needs and work with them to generate the event of their dreams.

Our customer's wishes turn out to be our instructions. We're certain that our business venture would be an achievement.

The Structure

Our website's structure is split in two big parts: **Backend** and **Frontend**.

For the Backend, I have chosen to use **Node.js** with **Firebase** as a Backend-as-a-Service – BaaS.

Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a web browser. (3) Node.js lets developers use JavaScript to write down command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts.

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014. As of March 2020, the Firebase platform has 19 products, which are used in more than 1.5 million apps including 9GAG. Out of this services we have used **Firebase Authentication**, **Firebase Realtime Database** and **Firebase Storage**. (4)

For the Frontend, I have chosen **React** with a well-liked framework called **Material UI** and a state management library called **Redux.**

React is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies. React makes it painless to create interactive UIs. Design simple views for every state in your application, and React will efficiently update and render just the proper components when your data changes. (5)

Material-UI is an open-source project that features React components that implement Google's Material Design. With over 35,000 stars on GitHub, Material-UI is one of the top user interface libraries for React out there. (6)

2. What does it take for the perfect wedding?

Wedding Booking is a website where we unite marriage service providers with couples that wants to get married. We can categorize the wedding day as the most precious day, but for that to be perfect a couple have to go through a long, stressful and endless journey of acquiring the preparations needed for the master day. They have to choose a location first.

Location

For many, the available spots are overwhelming. From churches to hotels to restaurants to castles, finding the perfect location can be quite a task. They have to pick a place that has the right size for the number of guests, is within budget, is available on the date that they want to marry and even have a large enough parking space.

Music

A lot of brides are very anxious about their wedding entrance song. All brides want that perfect wedding entrance song to walk down the aisle to. The songs during the whole wedding have to be also entertaining as much as possible for the guests. So, the music part is a matter of preference and should always mean something to the bride and groom. They still have to choose between DJ's or live formation or even both.

Photo & Video

Your wedding photo albums and wedding video will be the way you revisit some of the best moments of your wedding day. Don't let anyone tell you its worth skimping on wedding photography or videography. After the wedding day, those photos and video are all that's left to help you remember the day. Seems like a pretty significant set of vendors, right?

Entertainment

Nowadays, all the couples want to entertain and keep the guests satisfied during the wedding. They want to do that along with being in trend, so they hire a group of people to dance for them and book magic mirrors so they can make photos by themselves. The families of the couple surprise the bride and the groom with dry ice for the couple dancing part. So, all the wedding is about entertainment.

Others

For the wedding to be memorable, they also need photographers and video-cameras so they can rewatch the wedding and even show them to the future kids, but let’s not forget about the cake too. The palette of choosing is very vast for this also, but many couples don’t know the whole process of organizing the wedding by themselves, so they hire a wedding planner.

Wedding Booking is the place where we can find what do we need from the beginning to the end of the wedding day. For the wedding service providers, this is a good opportunity to promote and show to people what they have to offer on a website where couples should find anything.

2.1. Advantages of using Wedding Booking

We don’t need to say it again, wedding planning is rough stuff. And the hardest part of it all is the vendor selection. Can’t say ‘I Do’ without your dream team behind you.

- ✓ Time Saving
- ✓ Safety and Security
- ✓ Continuously updated information
- ✓ Easy to see the list of bookings
- ✓ Book a vendor anytime
- ✓ The comfort to book from home rather than looking for vendors all over the city
- ✓ All offers at one place
- ✓ Determine a budget → Targeted vendor audiences → Keeping within budget
- ✓ No pressure in taking a complex decision
- ✓ Everything ready in anytime of the day
- ✓ No appointments
- ✓ Opportunity to book everything in a day
- ✓ Easy wedding preparations for all couples
- ✓ Avoid unnecessary conflicts based on stressful decisions
- ✓ Compare offers
- ✓ Read comments, check likes and see how engaging a service is
- ✓ Perfect wedding at clicks ahead

3. Application Architecture

Having a good starting line when it comes to our project architecture is vital for the lifetime of the project itself and how you will be ready to tackle changing needs within the future. A bad, messy project architecture often leads to:

- Unreadable and messy code, making the development process longer and also the product itself harder to test
- Useless repetition, making code harder to maintain and manage
- Difficulty implementing new features. Since the structure can become a complete mess, adding a new feature without messing up existing code can become a true problem

With these points in mind, we can all agree that our project architecture is extremely important, and that we may declare some points which will help us determine what this architecture must help us do:

- ✓ Achieve clean and readable code
- ✓ Achieve reusable pieces of code across our application
- ✓ Help us to avoid repetitions
- ✓ Make life easier when adding a new feature into our application

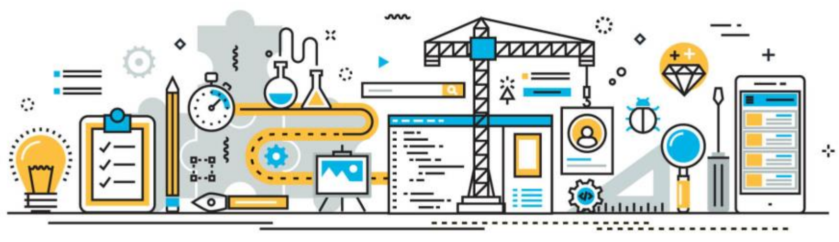


Figure 3.1. Website Architecture

3.1. BackEnd

The back-end is the code that runs on the server, that receives requests from the clients, and contains the logic to send the suitable data back to the client. The back-end also includes the database, which will persistently store all of the data for the application.

A server is just a computer that listens for incoming requests. Though there are machines made and optimized for this particular purpose, any computer that is connected to a network can act as a server.

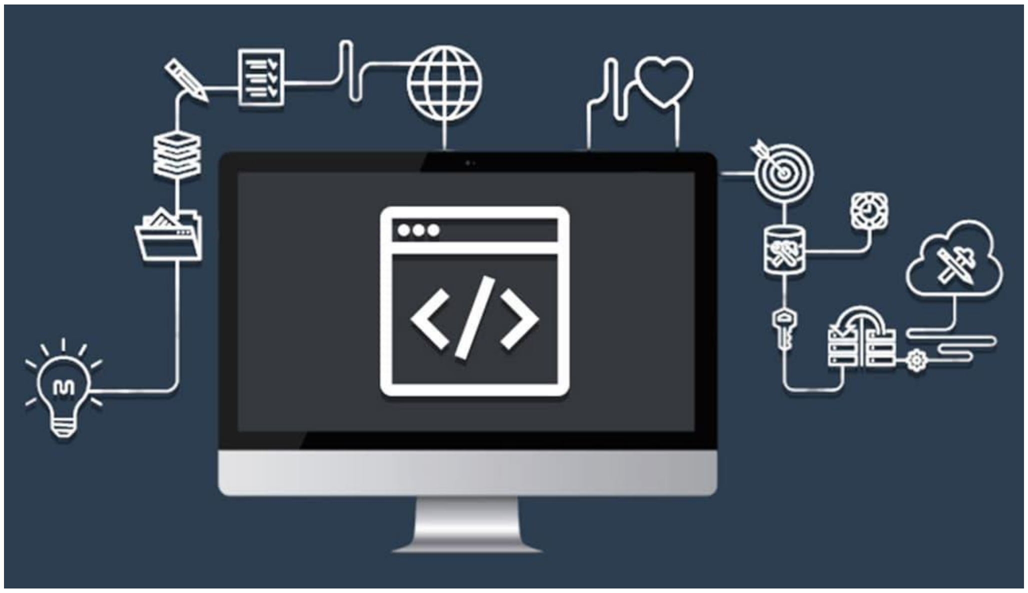


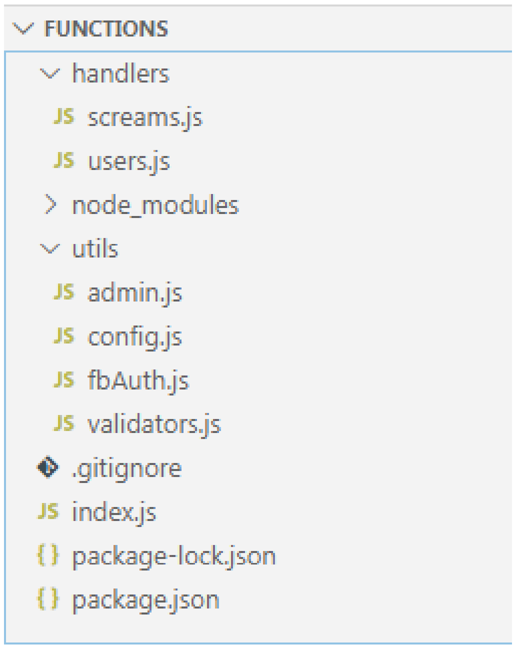
Figure 3.1.1. Backend

Node.JS

Figure 3.1.2. NodeJS

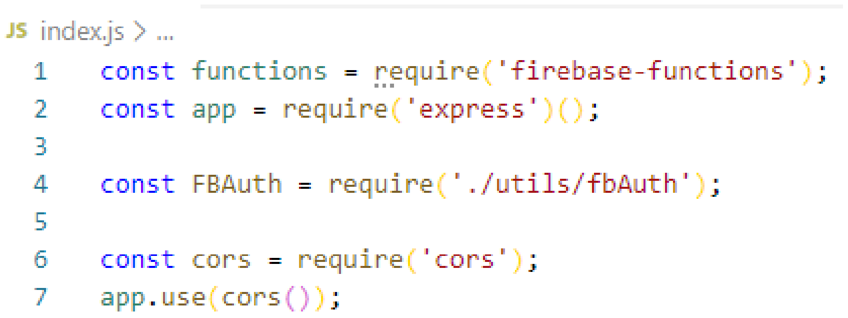
Node.js is an application runtime environment that allows you to write down server-side applications in JavaScript. Due to its unique I/O model, it excels at the type of scalable and real-time situations we are increasingly demanding of our servers. (7)

We have organized our folders and files as the image below. The functions folder contains all our files including the node_modules. In the handlers folder we have two files screams and users, where each file contains the implementation of each function called by the router. In the utils folder we have the configuration for the firebase along with the firebase admin SDK. The fbAuth is our middleware and in the validators file we validate the input data.



```
FUNCTIONS
  handlers
    screams.js
    users.js
  node_modules
  utils
    admin.js
    config.js
    fbAuth.js
    validators.js
  .gitignore
  index.js
  package-lock.json
  package.json
```

Figure 3.1.3. Architecture Tree



```
index.js > ...
const functions = require('firebase-functions');
const app = require('express')();

const FBAuth = require('./utils/fbAuth');

const cors = require('cors');
app.use(cors());
```

Figure 3.1.4. Packages used

The **firebase-functions** package provides an SDK for outlining Cloud Functions for Firebase. Cloud Functions is a hosted, private, and scalable Node.js environment where you will be able to run JavaScript code. The Firebase SDK for Cloud Functions integrates the Firebase platform by letting you write code that responds to events and invokes functionality exposed by other Firebase features. (8)

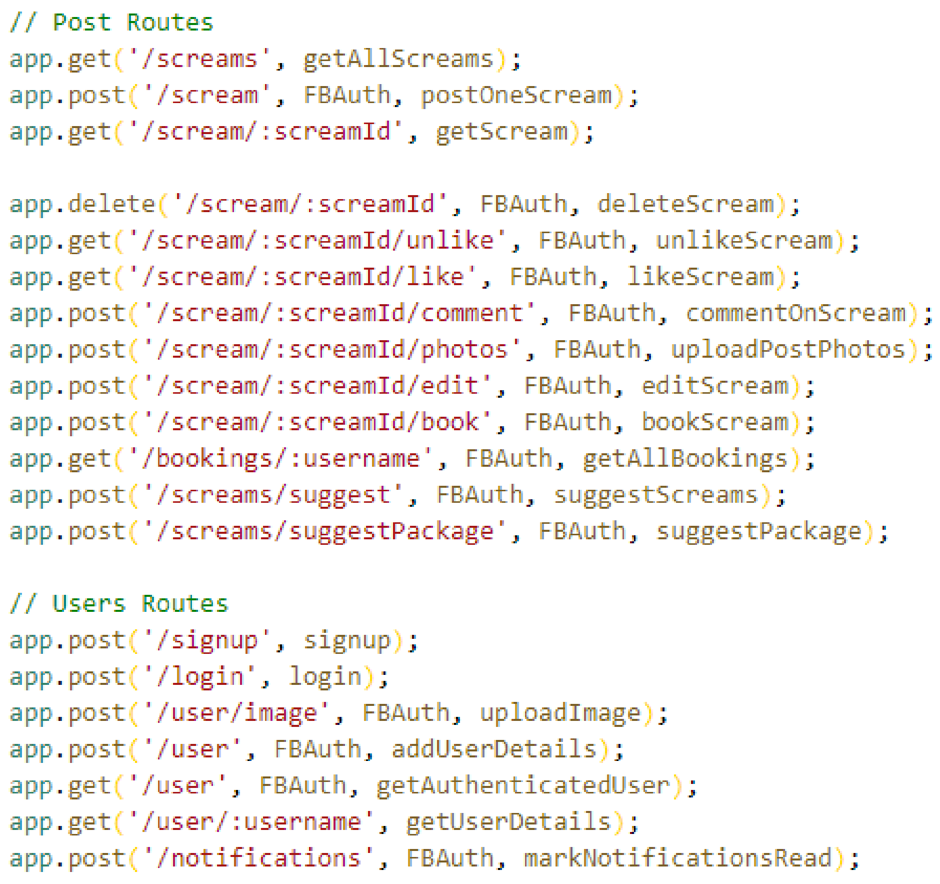
Express is a minimal, flexible and verasatile Node.js web application framework that provides a robust set of features for web and mobile applications. With a myriad of HTTP utility methods and middleware at your disposal, creating a robust API is quick and straightforward.

express

Figure 3.1.5. Express

The server runs an app that contains logic about how to respond to various requests based on the **HTTP** verb and therefore the Uniform Resource Identifier (**URI**). The pair of an HTTP verb and a URI is termed a **route** and matching them based on a request is called **routing**.

An **API** is a collection of clearly defined methods of communication between different software components. More specifically, a web API is the interface created by the back-end: the gathering of endpoints and the resources these endpoints expose. A web API is defined by the types of requests that it can handle, which is determined by the routes that it defines, and the kinds of responses that the clients can expect to receive after hitting those routes. One web API can be used to provide data for different front-ends.

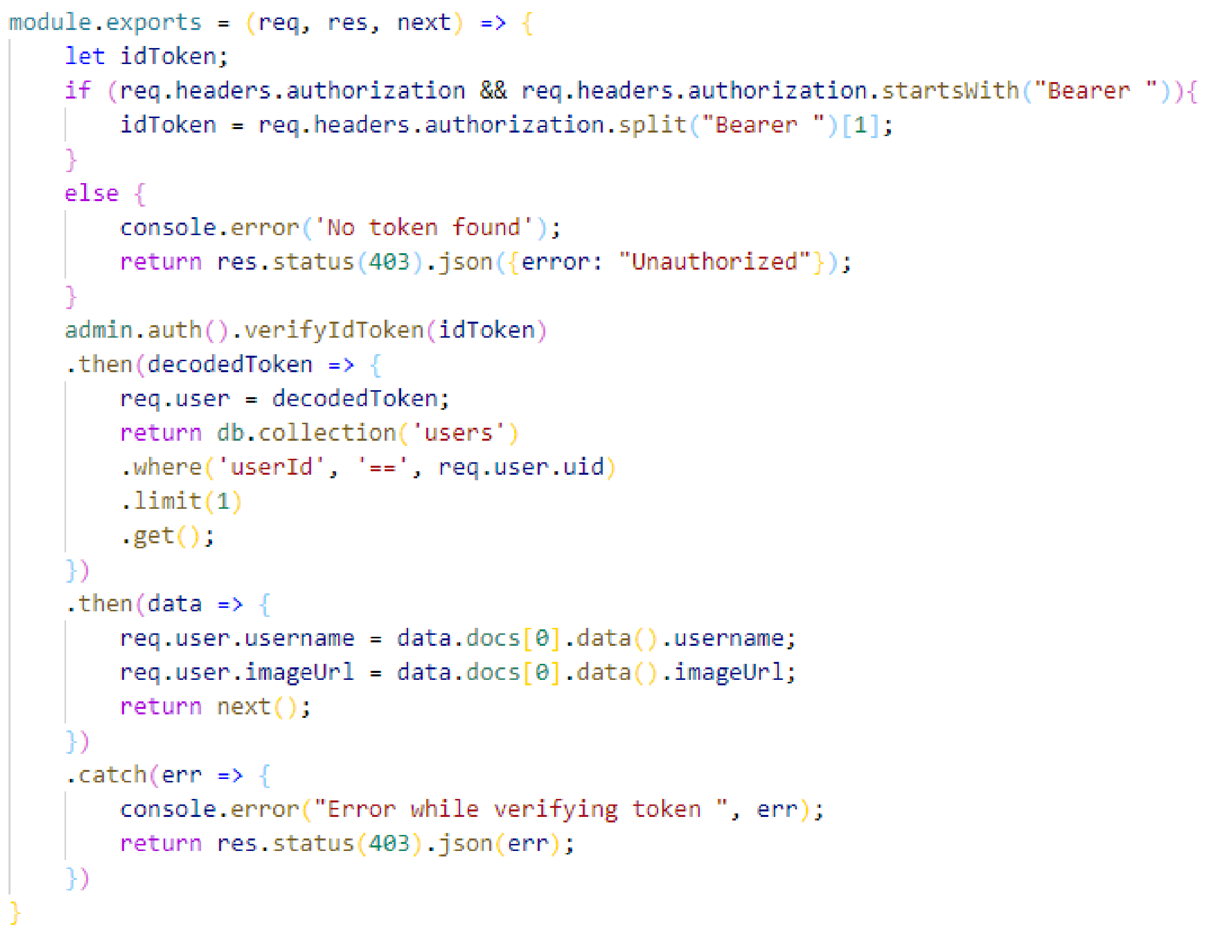


```
// Post Routes
app.get('/screams', getAllScreams);
app.post('/scream', FBAuth, postOneScream);
app.get('/scream/:screamId', getScream);

app.delete('/scream/:screamId', FBAuth, deleteScream);
app.get('/scream/:screamId/unlike', FBAuth, unlikeScream);
app.get('/scream/:screamId/like', FBAuth, likeScream);
app.post('/scream/:screamId/comment', FBAuth, commentOnScream);
app.post('/scream/:screamId/photos', FBAuth, uploadPostPhotos);
app.post('/scream/:screamId/edit', FBAuth, editScream);
app.post('/scream/:screamId/book', FBAuth, bookScream);
app.get('/bookings/:username', FBAuth, getAllBookings);
app.post('/screams/suggest', FBAuth, suggestScreams);
app.post('/screams/suggestPackage', FBAuth, suggestPackage);

// Users Routes
app.post('/signup', signup);
app.post('/login', login);
app.post('/user/image', FBAuth, uploadImage);
app.post('/user', FBAuth, addUserDetails);
app.get('/user', FBAuth, getAuthenticatedUser);
app.get('/user/:username', getUserDetails);
app.post('/notifications', FBAuth, markNotificationsRead);
```

Figure 3.1.6. Routes

In this context, **middleware** is any code that executes between the server receiving a request and sending a response. These middleware functions might modify the request object, query the database, or otherwise process the incoming request. Middleware functions typically end by passing control to the next middleware function, rather than by sending a response. Eventually, a middleware function will be called that ends the request-response cycle by sending an HTTP response back to the client.

Our middleware verifies if the request header contains a valid token, decode it, check for the specified user if exists and modifies the request object.



```
module.exports = (req, res, next) => {
    let idToken;
    if (req.headers.authorization && req.headers.authorization.startsWith("Bearer ")){
        idToken = req.headers.authorization.split("Bearer ")[1];
    }
    else {
        console.error('No token found');
        return res.status(403).json({error: "Unauthorized"});
    }
    admin.auth().verifyIdToken(idToken)
    .then(decodedToken => {
        req.user = decodedToken;
        return db.collection('users')
        .where('userId', '==', req.user.uid)
        .limit(1)
        .get();
    })
    .then(data => {
        req.user.username = data.docs[0].data().username;
        req.user.imageUrl = data.docs[0].data().imageUrl;
        return next();
    })
    .catch(err => {
        console.error("Error while verifying token ", err);
        return res.status(403).json(err);
    })
}
```

Figure 3.1.7. Middleware

Firebase

Figure 3.1.8. Firebase Logo

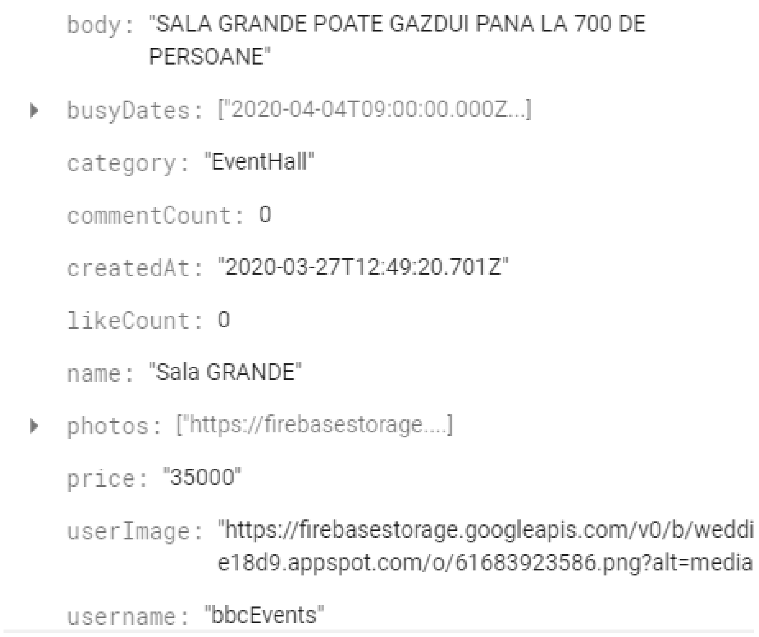
Firebase is **Google's** mobile platform that helps developers quickly build high-quality applications.

Firebase services used in this application:

- **Cloud Firestore** is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform. Like **Firebase Realtime Database**, it keeps your data in sync across client apps through realtime listeners and offers offline support for mobile and web so you will be able to build responsive apps that work regardless of network latency or Internet connectivity. Cloud Firestore also offers seamless integration with other Firebase and Google Cloud Platform products, including Cloud Functions. (9)

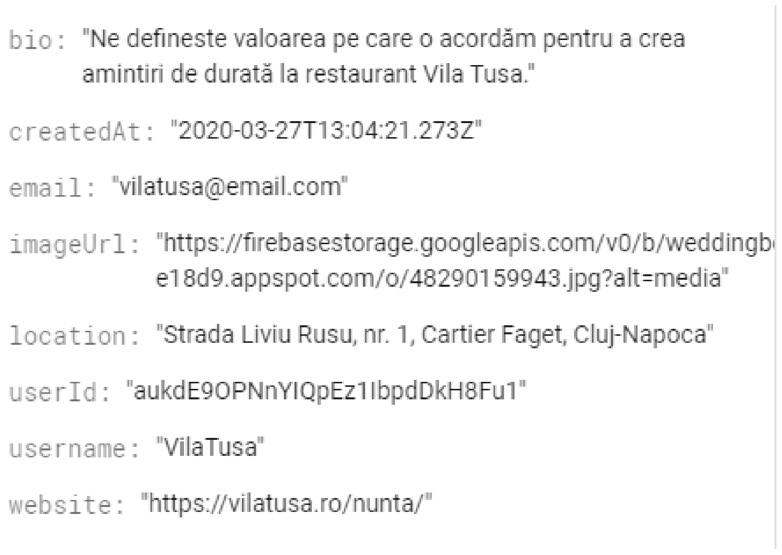
Figure 3.1.9. Cloud Firestore

In our **Cloud Firestore** we choose to save entities like: users, screams, notifications, likes, comments and bookings.



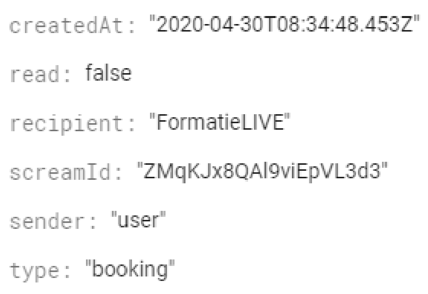
```
body: "SALA GRANDE POATE GAZDUI PANA LA 700 DE
       PERSOANE"
▸ busyDates: ["2020-04-04T09:00:00.000Z...]
category: "EventHall"
commentCount: 0
createdAt: "2020-03-27T12:49:20.701Z"
likeCount: 0
name: "Sala GRANDE"
▸ photos: ["https://firebasestorage....]
price: "35000"
userImage: "https://firebasestorage.googleapis.com/v0/b/weddi
            e18d9.appspot.com/o/61683923586.png?alt=media
username: "bbcEvents"
```

Figure 3.1.10. Users Entity



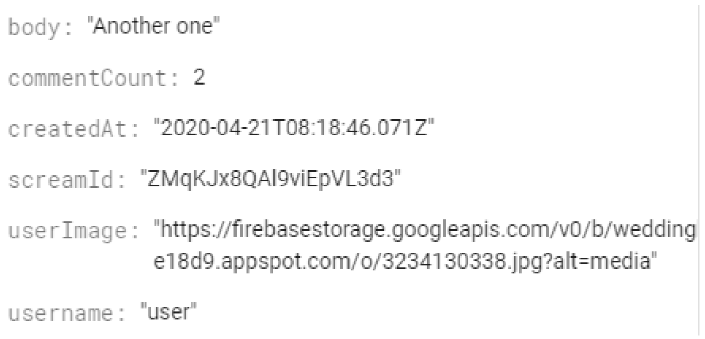
```
bio: "Ne defineste valoarea pe care o acordăm pentru a crea
      amintiri de durată la restaurant Vila Tusa."
createdAt: "2020-03-27T13:04:21.273Z"
email: "vilatusa@email.com"
imageUrl: "https://firebasestorage.googleapis.com/v0/b/weddingb
           e18d9.appspot.com/o/48290159943.jpg?alt=media"
location: "Strada Liviu Rusu, nr. 1, Cartier Faget, Cluj-Napoca"
userId: "aukdE9OPNnYIQpEz1IbpdDkH8Fu1"
username: "VilaTusa"
website: "https://vilatusa.ro/nunta/"
```

Figure 3.1.11. Screams Entity



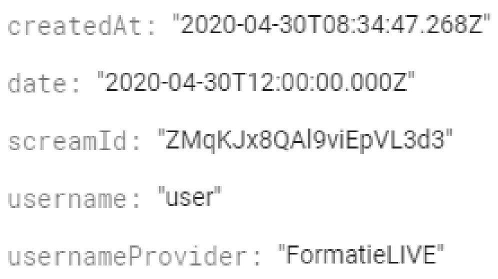
```
createdAt: "2020-04-30T08:34:48.453Z"
read: false
recipient: "FormatieLIVE"
screamId: "ZMqKJx8QAl9viEpVL3d3"
sender: "user"
type: "booking"
```

Figure 3.1.12. Notifications Table



```
body: "Another one"
commentCount: 2
createdAt: "2020-04-21T08:18:46.071Z"
screamId: "ZMqKJx8QAl9viEpVL3d3"
userImage: "https://firebasestorage.googleapis.com/v0/b/wedding
            e18d9.appspot.com/o/3234130338.jpg?alt=media"
username: "user"
```

Figure 3.1.13. Comments Table



```
createdAt: "2020-04-30T08:34:47.268Z"
date: "2020-04-30T12:00:00.000Z"
screamId: "ZMqKJx8QAl9viEpVL3d3"
username: "user"
usernameProvider: "FormatieLIVE"
```

Figure 3.1.14. Bookings Table

- **Firebase Storage** lets you upload and share user generated content, such as images and videos, which allows you to create rich media content into your apps. Firebase Storage stores this data in a Google Cloud Storage bucket, a petabyte scale object storage solution with high availability and global redundancy. Firebase Storage lets you securely upload these files directly from mobile devices and web browsers, handling spotty networks with ease. (9) Here we save the users profile pictures together with the images for promoting a service.

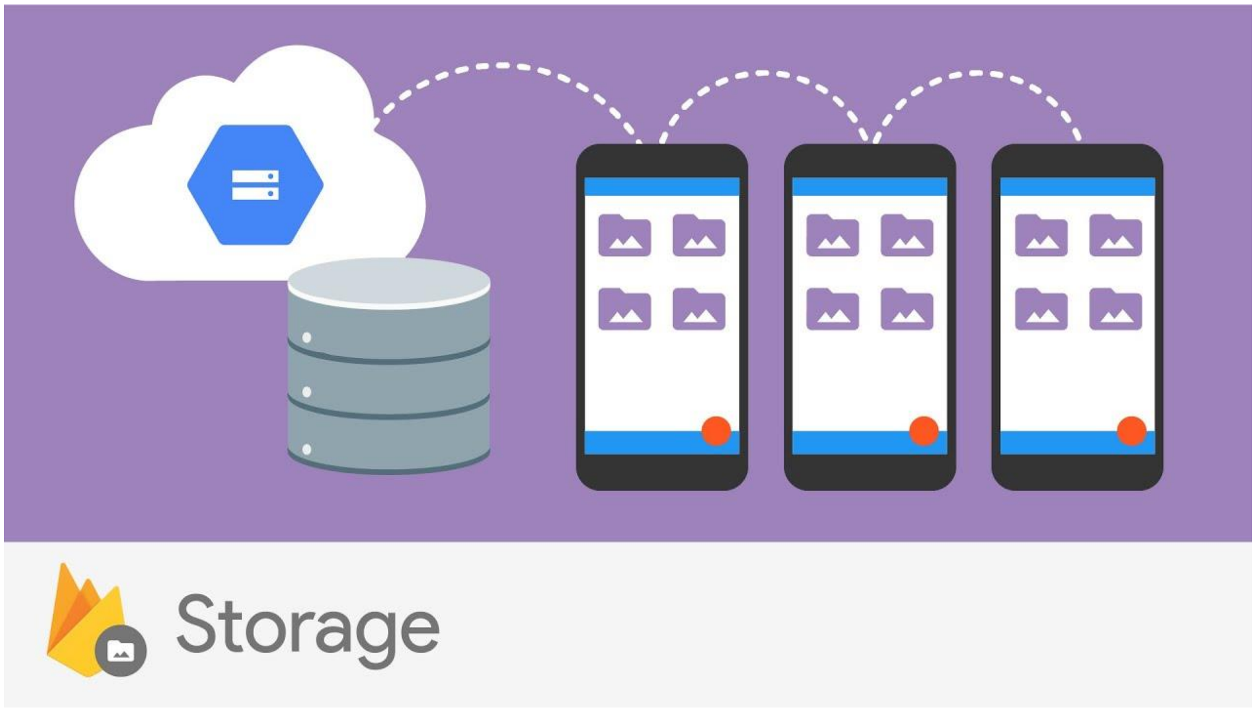


Figure 3.1.15. Firebase Storage

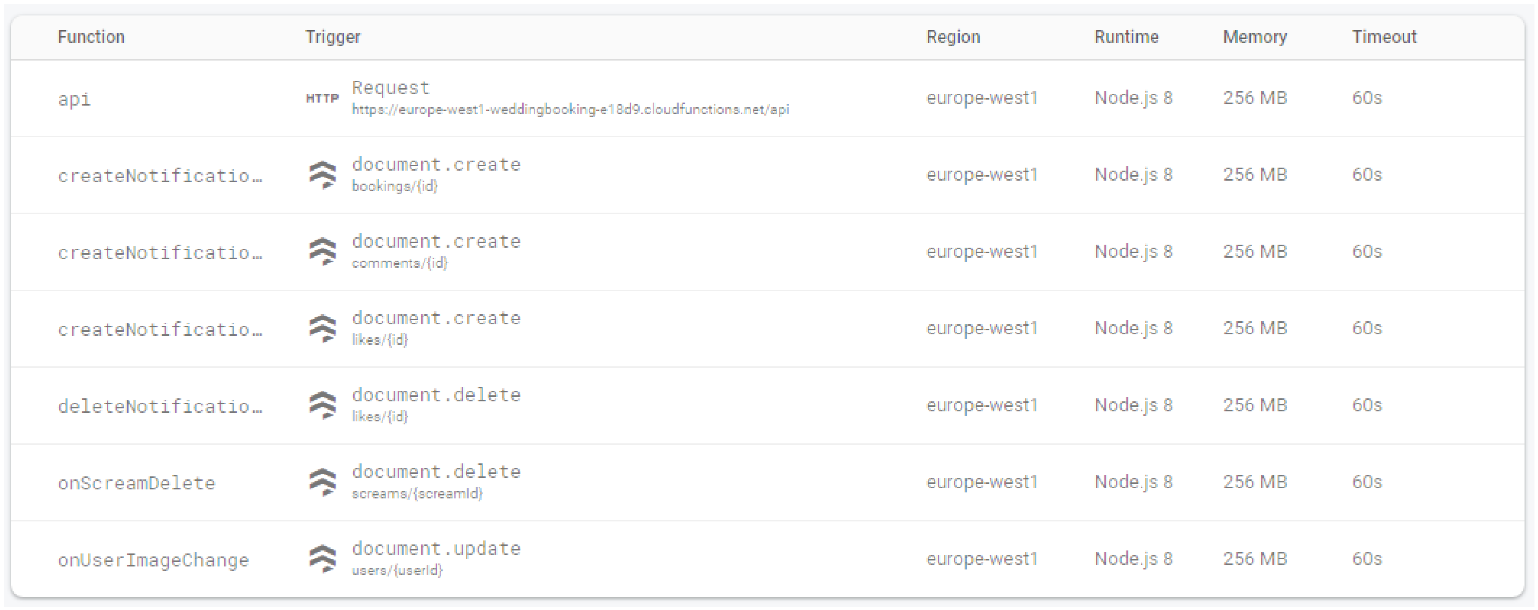
- **Firebase Authentication** provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more. Once a user authenticates, information about the user is returned to the device via callbacks this allows developers to personalize the application's user experience for that specific user. The user information contains a unique ID which is guaranteed to be distinct across all providers and never changing for specific authenticated user. This unique ID is used to identify your user and what parts of your back-end system they're authorized to access. Firebase will also manage your user session so that users will remain logged in after the browser of application restarts. (9) We have used this tool for registering users and logging them into their accounts on our website.

Firebase Authentication
Login
Firebase

Figure 3.1.16. Firebase Authentication

- **Cloud Functions** for Firebase is a serverless framework that enables you to automatically run backend code in response to events triggered by Firebase features and HTTPS requests. Your JavaScript or TypeScript code is stored in Google's cloud and runs in a managed environment. There's no need to manage and scale your own servers. (9)

Figure 3.1.17. Cloud Functions



Function	Trigger	Region	Runtime	Memory	Timeout
api	HTTP Request https://europe-west1-weddingbooking-e18d9.cloudfunctions.net/api	europe-west1	Node.js 8	256 MB	60s
createNotificatio…	document.create bookings/{id}	europe-west1	Node.js 8	256 MB	60s
createNotificatio…	document.create comments/{id}	europe-west1	Node.js 8	256 MB	60s
createNotificatio…	document.create likes/{id}	europe-west1	Node.js 8	256 MB	60s
deleteNotificatio…	document.delete likes/{id}	europe-west1	Node.js 8	256 MB	60s
onScreamDelete	document.delete screams/{screamId}	europe-west1	Node.js 8	256 MB	60s
onUserImageChange	document.update users/{userId}	europe-west1	Node.js 8	256 MB	60s

Figure 3.1.18. Functions with their triggers

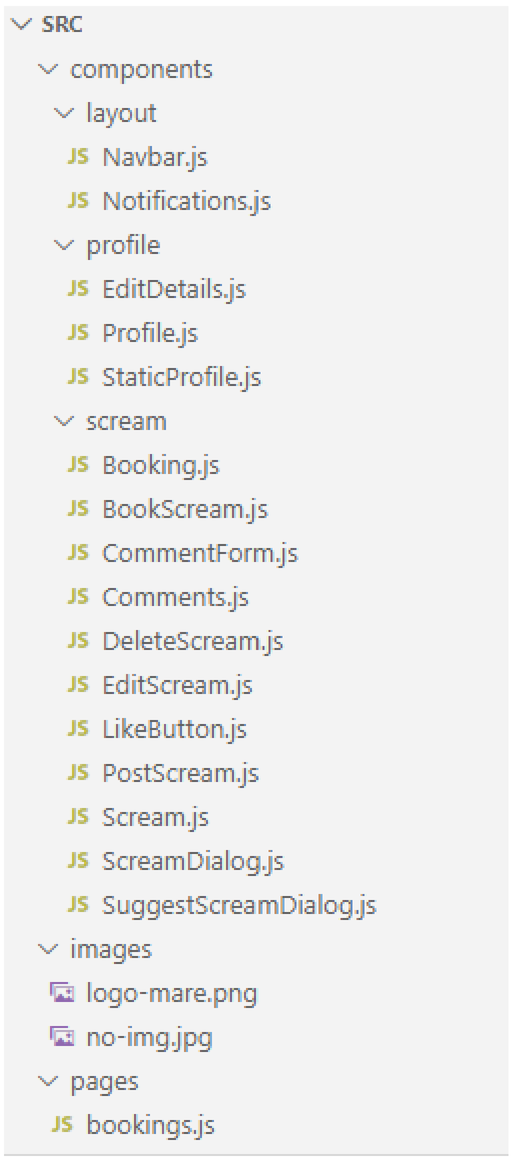
3.2. FrontEnd

Front-end web development is the practice of converting data to a graphical interface, through the use of HTML, CSS, and **JavaScript**, so that users can view and interact with that data.

React

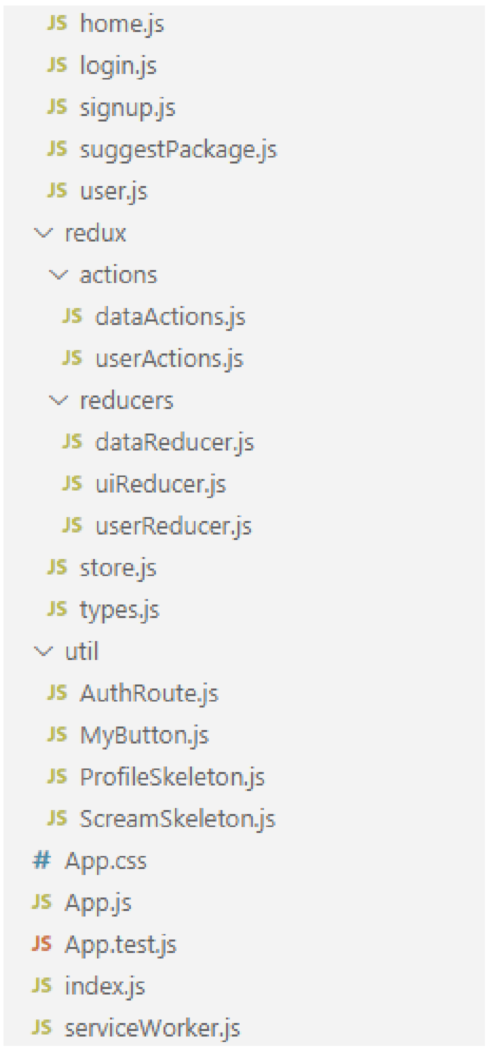
Figure 3.2.1. React Logo

React is a JavaScript library for building user interfaces. It is maintained by **Facebook** and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with rendering data to the DOM, and so creating React applications usually requires the use of additional libraries for state management and routing. **Redux** and **React Router** are respective examples of such libraries. (5)



```
SRC
  components
    layout
      Navbar.js
      Notifications.js
    profile
      EditDetails.js
      Profile.js
      StaticProfile.js
    scream
      Booking.js
      BookScream.js
      CommentForm.js
      Comments.js
      DeleteScream.js
      EditScream.js
      LikeButton.js
      PostScream.js
      Scream.js
      ScreamDialog.js
      SuggestScreamDialog.js
  images
    logo-mare.png
    no-img.jpg
  pages
    bookings.js
```

Figure 3.2.2. React Architecture



```
    home.js
    login.js
    signup.js
    suggestPackage.js
    user.js
  redux
    actions
      dataActions.js
      userActions.js
    reducers
      dataReducer.js
      uiReducer.js
      userReducer.js
    store.js
    types.js
  util
    AuthRoute.js
    MyButton.js
    ProfileSkeleton.js
    ScreamSkeleton.js
  App.css
  App.js
  App.test.js
  index.js
  serviceWorker.js
```

Figure 3.2.3. React Architecture

Our React Application is divided in 5 folders: components, pages, redux, util and images.

In the **components folder** we build and structured each component so that we can use and reuse them whenever needed. This is splitted in :

- **Layout Part** where we have the Navigation Menu component along with the notifications. Here we link each icon from the navbar with its component. The component being rendered whenever an icon is clicked. We have two navigation menus for a logged in user and for a not register one. For the logged in user, the navigation is also render regarding the user type: Service Provider or Married to be. In the notification component we have 3 types of notifications: Like, Comment or Booking. There we also have the time: how much time ago the action was taken.

- **Profile Part** where we have the profile itself along with the component for editing it and staticProfile where we have a non-editable profile. In the user box we show the credentials that he provided along with buttons for changing profile picture, editing details or the option for the user to logout. On the edit part we can change the bio, location website and phone.

- **Posts Part** where we have all the components related to the post and the actions that can be taken regarding its user type. A service provider can create, edit and delete a post but also engage with it. A married to be, has the ability to search for posts with different criterias, see post details, engage with them and search for a specific package with a specific date, budget and vendors categories.

In the **images** folder we keep only the Wedding Booking logo so we can use it in the pages that we want.

In the **pages** folder we maintain and store every page from our application. In the **Signup** page we have textfields to be completed for a user to setup an account. If any of this is violated we display error messages regarding the violation (Unique email and username). In the **Login** page we have to use the credentials created before. We will also validate this input data. In the **Home** page we display all the screams that corresponds to the criterias that a user select. At first the home page contains all the posts, but then the user can select to see posts that are available for a date, are within their budget and can select them regarding the categories the user needs. If a user clicks on a user name it redirects him to the users page where it can see all the posts that are related to that user, that is the **User** page.

The **Suggest Package** page is about our engagement algorithm, where a user can search for an entire package for the wedding with different criterias. There is also an option for “Suggested” where we select the services for you regarding the engagement each service have. From there you can book everything you need with a touch of a button.

After the user booked some services, those can be found at the **Bookings** page. There the post will be sorted ascending by date, so the vendors can check what comes first.

Redux

Figure 3.2.4. Redux Logo

Redux is an open-source JavaScript library for managing application state. It is most commonly used with libraries such as React or Angular for building user interfaces. Similar **Facebook's Flux architecture**, it was created by Dan Abramov and Andrew Clark.

Redux is a small library with a simple, limited API designed to be a predictable container for application state. It operates in a similar fashion to a reducing function, a functional programming concept. It is influenced by the functional programming language Elm. (10)

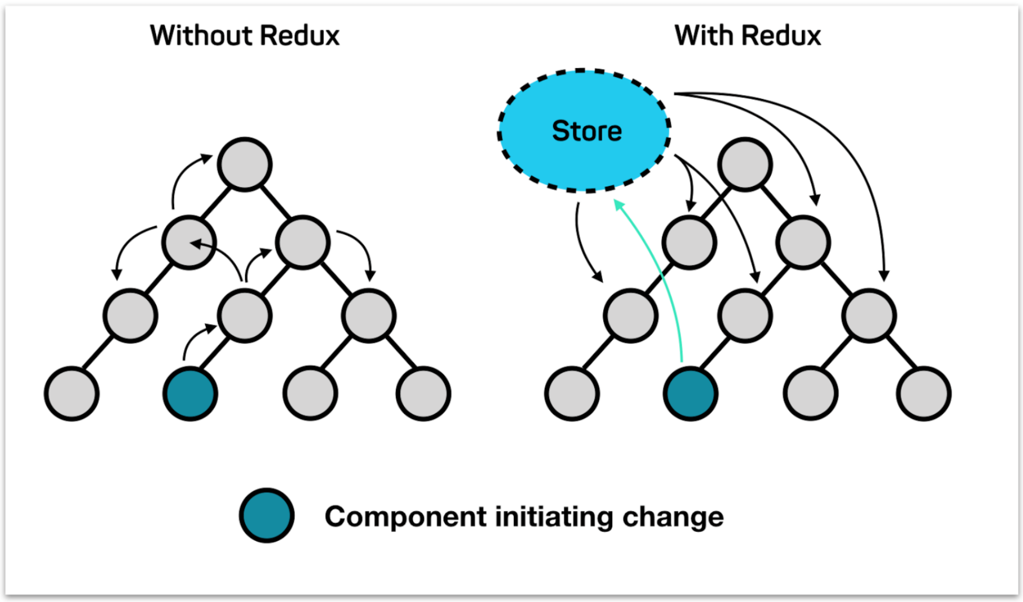
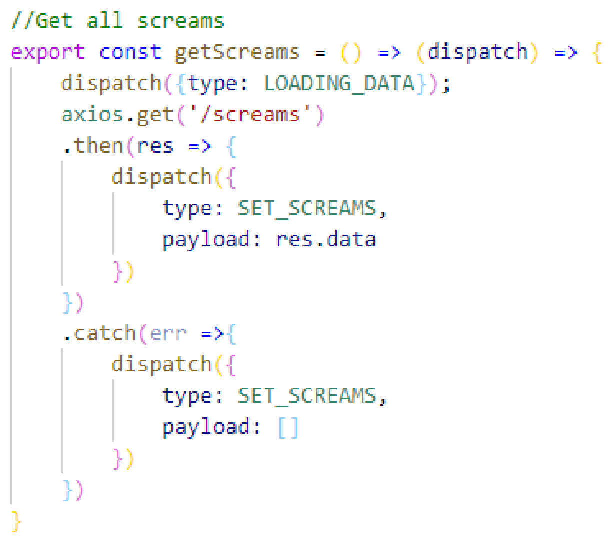


Figure 3.2.5. Redux vs Non-Redux

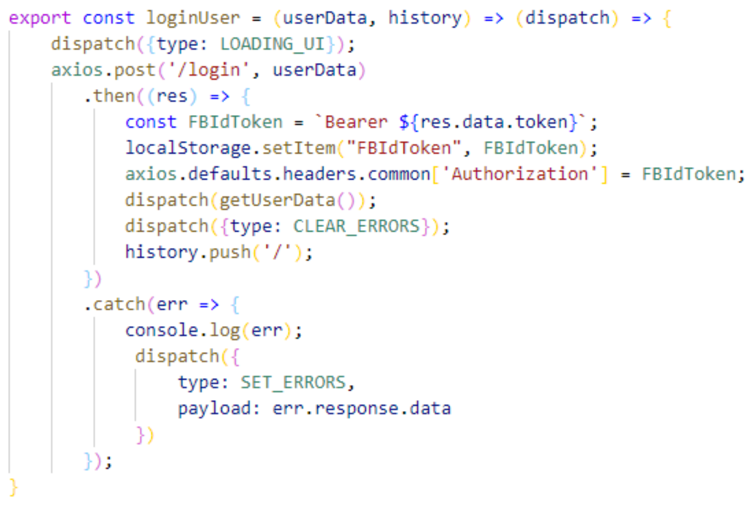
In the **redux** folder, we have three types of reducers:



```
//Get all screams
export const getScreams = () => (dispatch) => {
    dispatch({type: LOADING_DATA});
    axios.get('/screams')
    .then(res => {
        dispatch({
            type: SET_SCREAMS,
            payload: res.data
        })
    })
    .catch(err =>{
        dispatch({
            type: SET_SCREAMS,
            payload: []
        })
    })
}
```

- **Data reducer** who manage all the information related to posts : likes, unlikes, comments, read, create, edit, delete, book as well as loading data, so we can manage changes in the frontend. In the **Data actions** the requests are sent with axios and then we dispatch actions with a type and a payload. Types are defined in the types file.
-

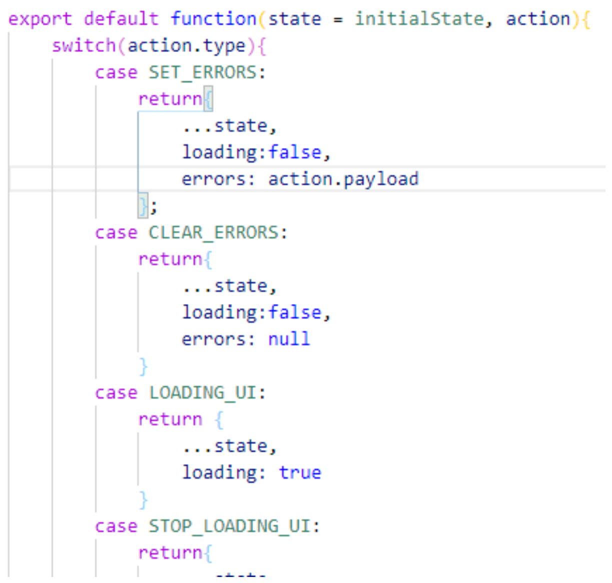
Figure 3.2.6. getScream Action dispatches SET_SCREAMS



```
export const loginUser = (userData, history) => (dispatch) => {
    dispatch({type: LOADING_UI});
    axios.post('/login', userData)
        .then((res) => {
            const FBIdToken = `Bearer ${res.data.token}`;
            localStorage.setItem("FBIdToken", FBIdToken);
            axios.defaults.headers.common['Authorization'] = FBIdToken;
            dispatch(getUserData());
            dispatch({type: CLEAR_ERRORS});
            history.push('/');
        })
        .catch(err => {
            console.log(err);
             dispatch({
                type: SET_ERRORS,
                payload: err.response.data
             })
        });
}
```

- **User reducer** manage the information and details about the authenticated user: credentials, notifications, comments, bookings, errors and loading user.
  In the **User actions** the requests are also sent with axios but here, despite dispatching some actions we are also setting the token and the authorization for the specific user.

Figure 3.2.7. loginUser action



```
export default function(state = initialState, action){
    switch(action.type){
        case SET_ERRORS:
            return{
                ...state,
                loading:false,
                errors: action.payload
            };
        case CLEAR_ERRORS:
            return{
                ...state,
                loading:false,
                errors: null
            }
        case LOADING_UI:
            return {
                ...state,
                loading: true
            }
        case STOP_LOADING_UI:
            return{
```

Figure 3.2.8. UI Reducer

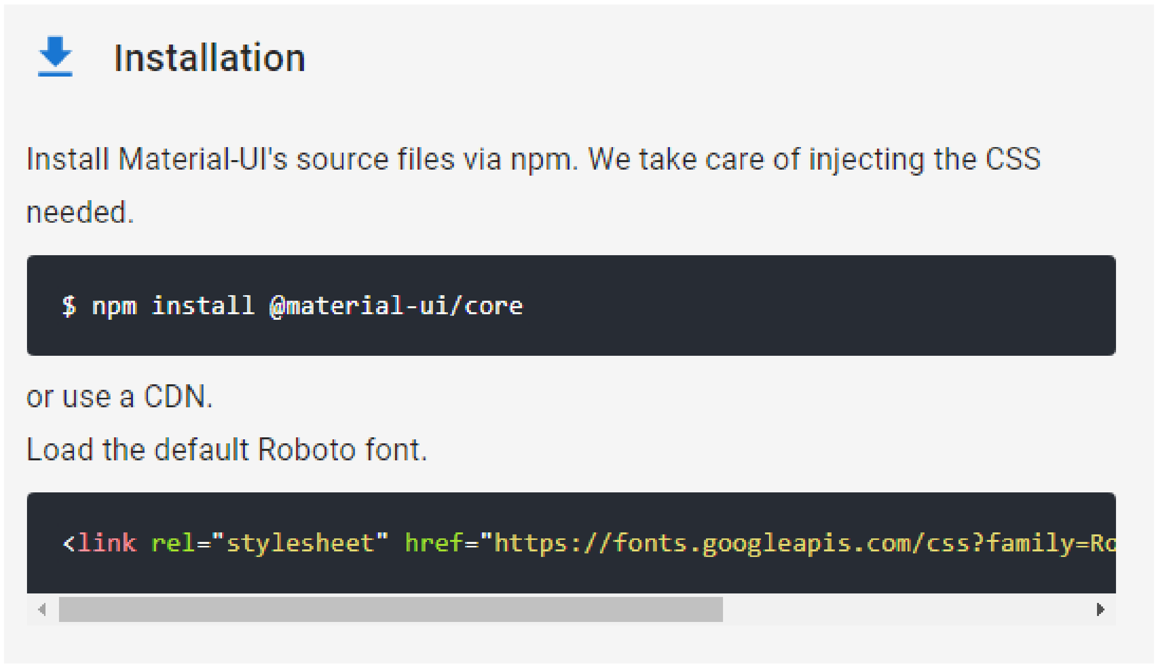
- And **UI reducer** where we mostly manage the errors and loadings from the states.

In the **util** folder we created two skeletons which will be displayed in the loading time for a smoother transition: **Profile Skeleton** and **Scream Skeleton.** Along with this skeletons we have created a Tooltip with a Button so we can use it multiple times without rewriting cod. The Auth Route component checks for the user to be authenticated and redirect him on the home page. This component is used for the Login and Signup page.

Material – UI Framework

Figure 3.2.9. Material Ui Logo

React components for faster and easier web development. Build your own design system, or start with Material Design. (11)



Installation

Install Material-UI's source files via npm. We take care of injecting the CSS needed.

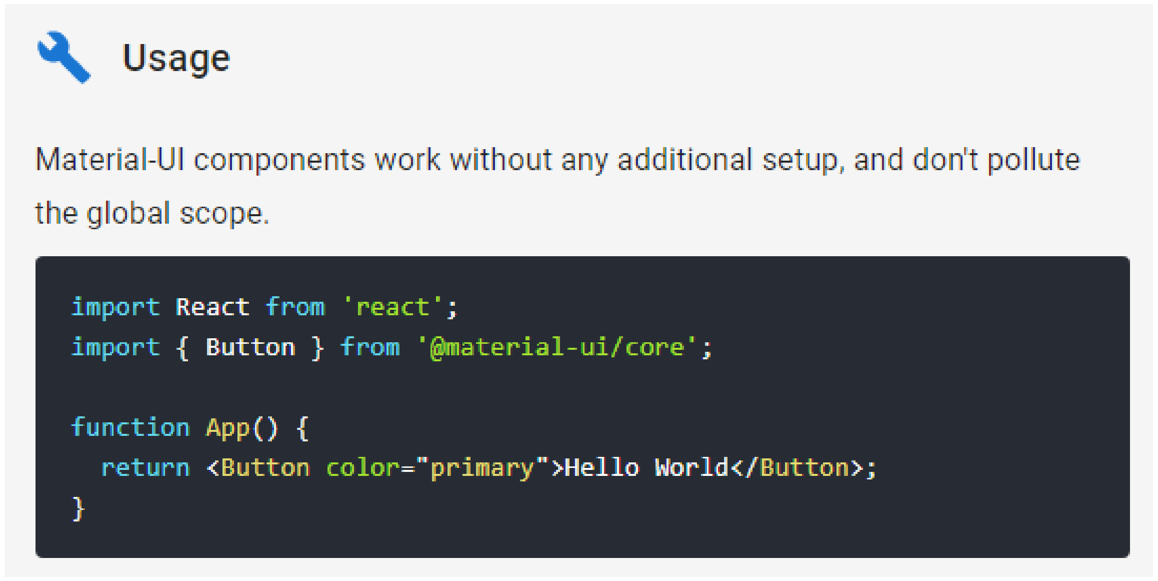
```
$ npm install @material-ui/core
```

or use a CDN.

Load the default Roboto font.

```
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Ro
```

Figure 3.2.10. Material – UI Installation



Usage

Material-UI components work without any additional setup, and don't pollute the global scope.

```
import React from 'react';
import { Button } from '@material-ui/core';

function App() {
  return <Button color="primary">Hello World</Button>;
}
```

Figure 3.2.11. Material – UI Usage

4. Engagement Service Suggestion Algorithm

Figure 4.1. Engagement Algorithm

Is the term “Engagement Algorithm” foreign to you? Engagement refers to any action a user takes with a post : commenting, liking, or booking it for example. An interesting note about the engagement algorithm that many don’t realize : reacting to a post in any way will increase the post engagement with it! As **Facebook** and **Instagram** use it we have tried to integrate our own version into the Wedding Booking Application.

Quality over Quantity

- For a Vendor, the best way to attract new people to book their services is that the people have to react at what he provides by liking or commenting. The post quality score will increase with the number of likes and reviews to it. Those adds up to the number of bookings the service have.

- For a Married to be, our algorithm suggest the best offers to take for a specific date, budget and multiple categories. This algorithm takes the input data and retrieves the offers that have the higher quality factor based on engagement : likes, comments and bookings.

Suggested Weight Algorithm

Our algorithm takes as input a date, several categories that we wish the package contain and a price. If the price is not present we take it as an unlimited budget.

- ✓ With the data that we have, we then move forward to selecting posts that are available for the specific date, are within budget and it is from a selected category;
- ✓ Then, for each category we make a set with the available posts;
- ✓ We create the cartesian combinations so we can take one element from each set
- ✓ For each combination we check if it is within budget, if yes we go to the next steps

For the last part, the user also selected a **Low**, **High** or **Suggested Priority:**

- For the **low** priority we calculate two packages (if possible) by calculating two different minimum packages that two services never intersect
- We do the same for the **high** priority by calculating the maximum packages
- For the **suggested** priority we have to apply the algorithm. We take all the bookings, likes and comments of the service and we calculate the quality factor:

$$\boldsymbol{Weight(S) = 25\% * Likes_{Count(S)} + 25\% * Comments_{Count(S)} + 50\% * Bookings_{Count(S)}}$$

S = the service post for whom will be calculated its quality;

Weight(S) = quality factor to be used in choosing “the best candidates”;

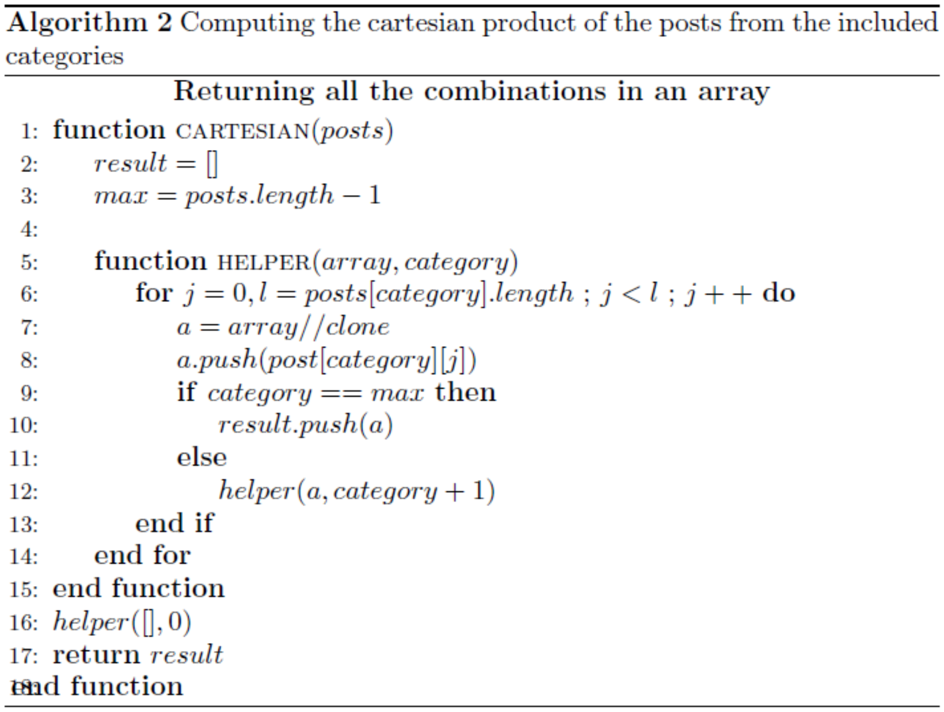
Likes_Count(S) = the number of likes of the post;

Comments_Count(S) = the number of comments of the post;

Bookings_Count(S) = how many times this service was booked;

After this factor is calculated for each service and for each combination we take the most two quality services if exist, else we take only the “best fit”.

Pseudocode Algorithm



Algorithm 2 Computing the cartesian product of the posts from the included categories

Returning all the combinations in an array

```
function CARTESIAN($posts$)
    $result = []$
    $max = posts.length - 1$

    function HELPER($array, category$)
        for $j = 0, l = posts[category].length$ ; $j < l$ ; $j++$ do
            $a = array//clone$
            $a.push(post[category][j])$
            if $category == max$ then
                $result.push(a)$
            else
                $helper(a, category + 1)$
            end if
        end for
    end function
    $helper([], 0)$
    return $result$
end function
```

The CARTESIAN function returns all combinations of the posts from the included categories. So, we have an empty array (result) where we will put all the combinations. The HELPER function takes as parameter an array and a category. Here we construct all combinations one by one at a time recursively and after that we add them in the result array (when we reach the number of elements in a combination equal to the number of categories).

Algorithm 1 Finding packages for specific criterias

Finds two packages that obey the date, budget, categories and type

```
function SUGGESTPACKAGE($date, budget, categories, type, posts, bookings$)
    for $post$ in $posts$ do
        if $checkDate(post, date), checkPrice(post, price), checkCategory(post, categories)$ then
            $goodPosts.push(post)$
        end if
    end for

    $allCombinations = cartesian(goodPosts)$;
    for $combination$ in $allCombinations$ do
        if $checkCombinationPrice(combination, price)$ then
            $inPriceCombinations.push(combination)$
        end if
    end for

    $costs = calculateCosts(inPriceCombinations)$
    if $type == 'low'$ then
        $index1, index2 = calculateTwoMin(costs, inPriceCombinations)$
    else
        if $type == 'high'$ then
            $index1, index2 = calculateTwoMax(costs, inPriceCombinations)$
        else
            if $type == 'suggested'$ then
                $qualityCoeff = calculateCoeff(inPriceCombinations, bookings)$
                $index1, index2 = calculateTwoMax(qualityCoeff, inPriceCombinations)$
            end if
        end if
    end if
    return $inPriceCombinations[index1], inPriceCombinations[index2]$;
end function
```

The SUGGESTPACKAGE function return if possible two packages for the input criterias. If one package can be constructed, we return it. For each input criteria we check if a post is a good one or a bad one. If it obeys the criterias we add it to the goodPosts array. Then with the cartesian function we compute all combinations from the good posts. After that, we check if each combination is within budget and add it to a inPrice array. For all inPrice combinations we calculate the cost of each combination and return two of them corresponding the input type of the user.

CalculateTwoMin takes as parameters the array which contain the costs of each combination and the array of all the combinations and return two indexes of minimum packages if possible, one if not.

CalculateTwoMax takes the same parameters but return two indexes of maximum packages if possible, one if not.

CalculateCoeff takes as parameters the array containing all in price combinations and the bookings, and return two indexes of the best valued posts if possible, one if not.

5. Design Patterns

In software engineering, a **design pattern** is a general repeatable solution to a commonly occurring problem in software design. A design pattern isn't a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations.

Design patterns can speed up the development process by providing tested, proven development paradigms. Effective software design requires considering issues that may not become visible until later in the implementation. Reusing design patterns helps to prevent subtle issues that can cause major problems and improves code readability for coders and architects familiar with the patterns. (12)

5.1. Creational Patterns

Creational patterns provide various object creation mechanisms, which increase flexibility and reuse of existing code.

Factory

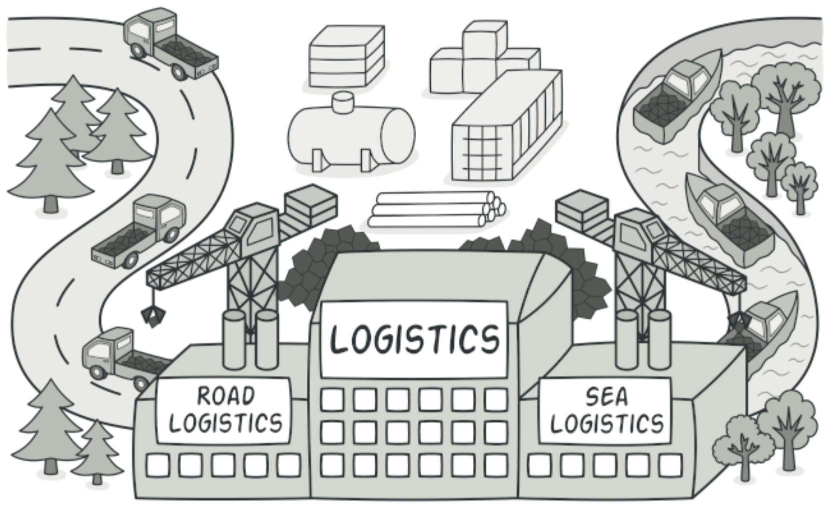


Figure 5.1.1. Factory Pattern

The Factory Method Pattern is a creational pattern that uses factory methods to deal with the problem of creating objects without having to specify the exact class of the object that will be created.

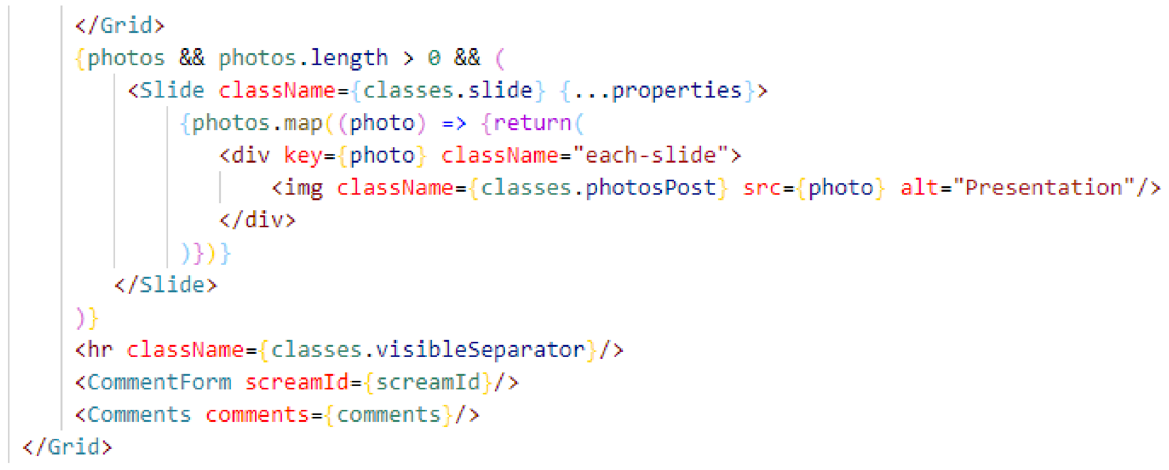
React-Slideshow Component

A simple slideshow component built with react that supports slide, fade and zoom effects.

Installation

npm i react-slideshow-image

Usage

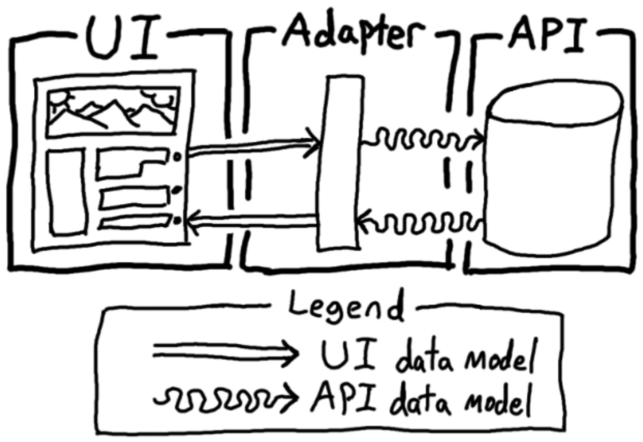


```
        </Grid>
        {photos && photos.length > 0 && (
            <Slide className={classes.slide} {...properties}>
                {photos.map((photo) => {return(
                    <div key={photo} className="each-slide">
                        <img className={classes.photosPost} src={photo} alt="Presentation"/>
                    </div>
                )})}
            </Slide>
        )}
        <hr className={classes.visibleSeparator}/>
        <CommentForm screamId={screamId}/>
        <Comments comments={comments}/>
    </Grid>
```

Figure 5.1.2. Slide as Factory Pattern

5.2. Structural Patterns

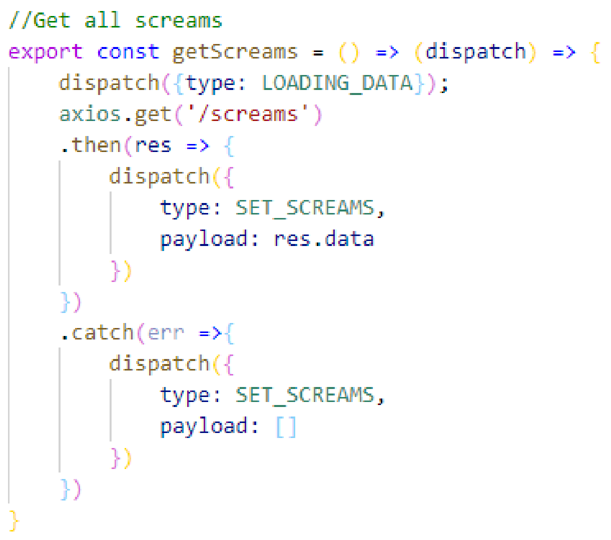
Structural patterns explain how to assemble objects and classes into larger structures while keeping these structures flexible and efficient.



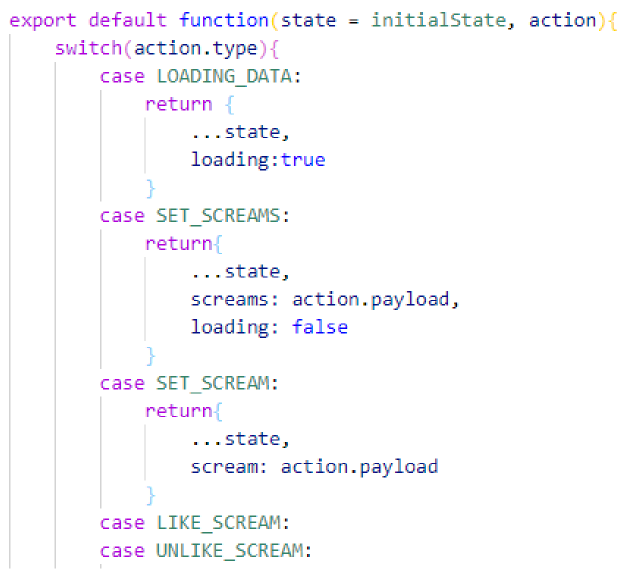
Adapter

Figure 5.2.1. Adapter Pattern

Convert the interface of a class into another interface clients expect. Adapter lets classes work together that couldn't otherwise because of incompatible interfaces.



```
//Get all screams
export const getScreams = () => (dispatch) => {
    dispatch({type: LOADING_DATA});
    axios.get('/screams')
    .then(res => {
        dispatch({
            type: SET_SCREAMS,
            payload: res.data
        })
    })
    .catch(err =>{
        dispatch({
            type: SET_SCREAMS,
            payload: []
        })
    })
}
```



```
export default function(state = initialState, action){
    switch(action.type){
        case LOADING_DATA:
            return {
                ...state,
                loading:true
            }
        case SET_SCREAMS:
            return{
                ...state,
                screams: action.payload,
                loading: false
            }
        case SET_SCREAM:
            return{
                ...state,
                scream: action.payload
            }
        case LIKE_SCREAM:
        case UNLIKE_SCREAM:
```

Figure 5.2.2. Adapter Pattern with Redux

5.3. Behavioral Patterns

Behavioral design patterns are concerned with algorithms and the assignment of responsibilities between objects.

Observer

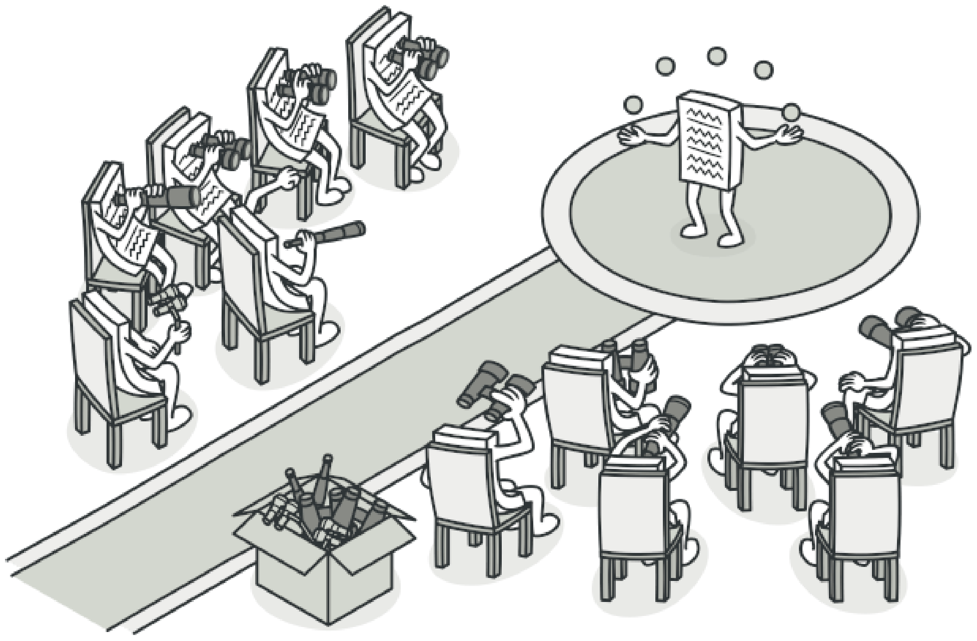
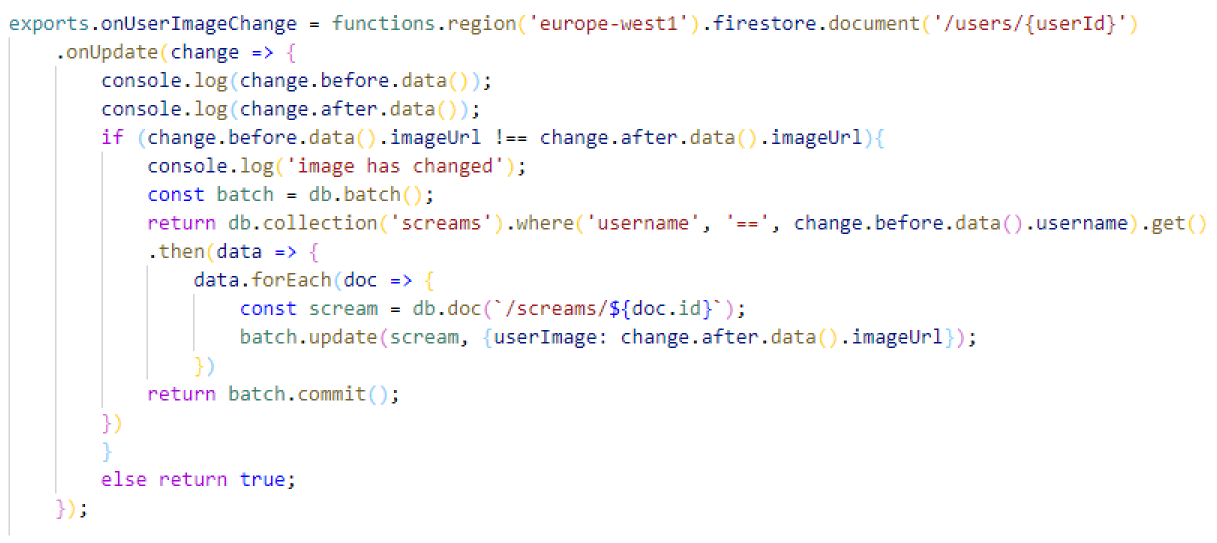


Figure 5.3.1. Observer Pattern

The Observer pattern offers a subscription model in which objects subscribe to an event and get notified when the event occurs. This pattern is the cornerstone of event driven programming, including JavaScript. The Observer pattern facilitates good object-oriented design and promotes loose coupling.



```
exports.onUserImageChange = functions.region('europe-west1').firestore.document('/users/{userId}')
    .onUpdate(change => {
        console.log(change.before.data());
        console.log(change.after.data());
        if (change.before.data().imageUrl !== change.after.data().imageUrl){
            console.log('image has changed');
            const batch = db.batch();
            return db.collection('screams').where('username', '==', change.before.data().username).get()
            .then(data => {
                data.forEach(doc => {
                    const scream = db.doc(`/screams/${doc.id}`);
                    batch.update(scream, {userImage: change.after.data().imageUrl});
                })
            return batch.commit();
        })
        }
        else return true;
    });
```

Figure 5.3.2. Observer on a Profile picture change

Iterator

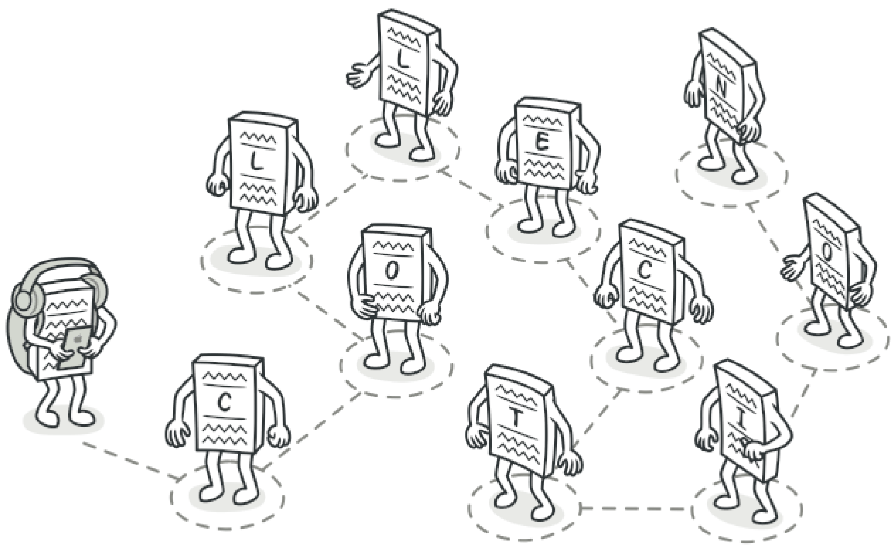


Figure 5.3.3. Iterator Pattern

Iterator is a behavioral design pattern that lets you traverse elements of a collection without exposing its underlying representation (list, stack, tree, etc.).

The **Slider** used on factory pattern contains an Iterator.

Methods

- **goNext()** It is used to programmatically transition the slide to the next one.
- **goBack()** If you want to show the previous slide, then use this function

- **goTo(index)** It can be used to transition the slide to a particular index.

State

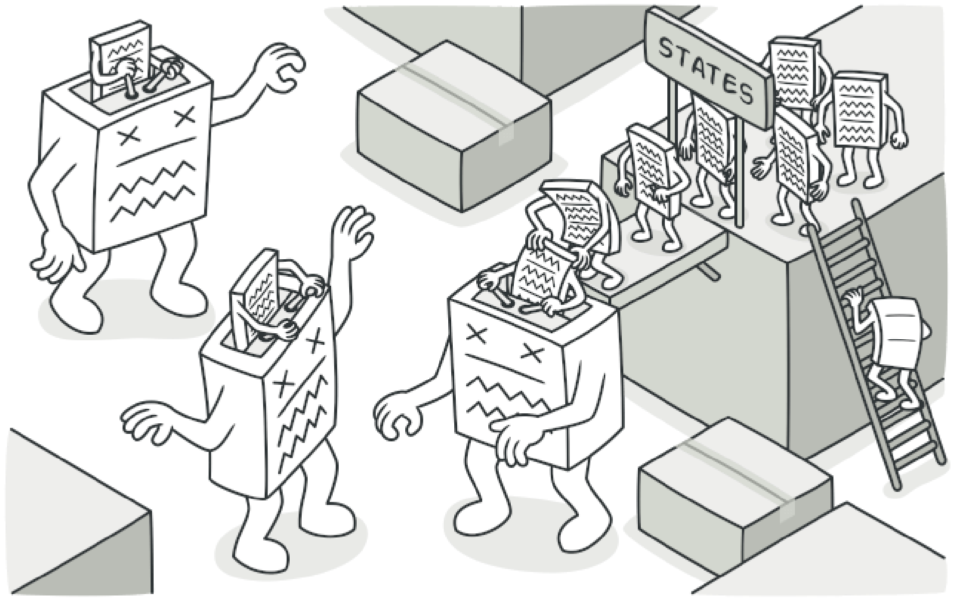
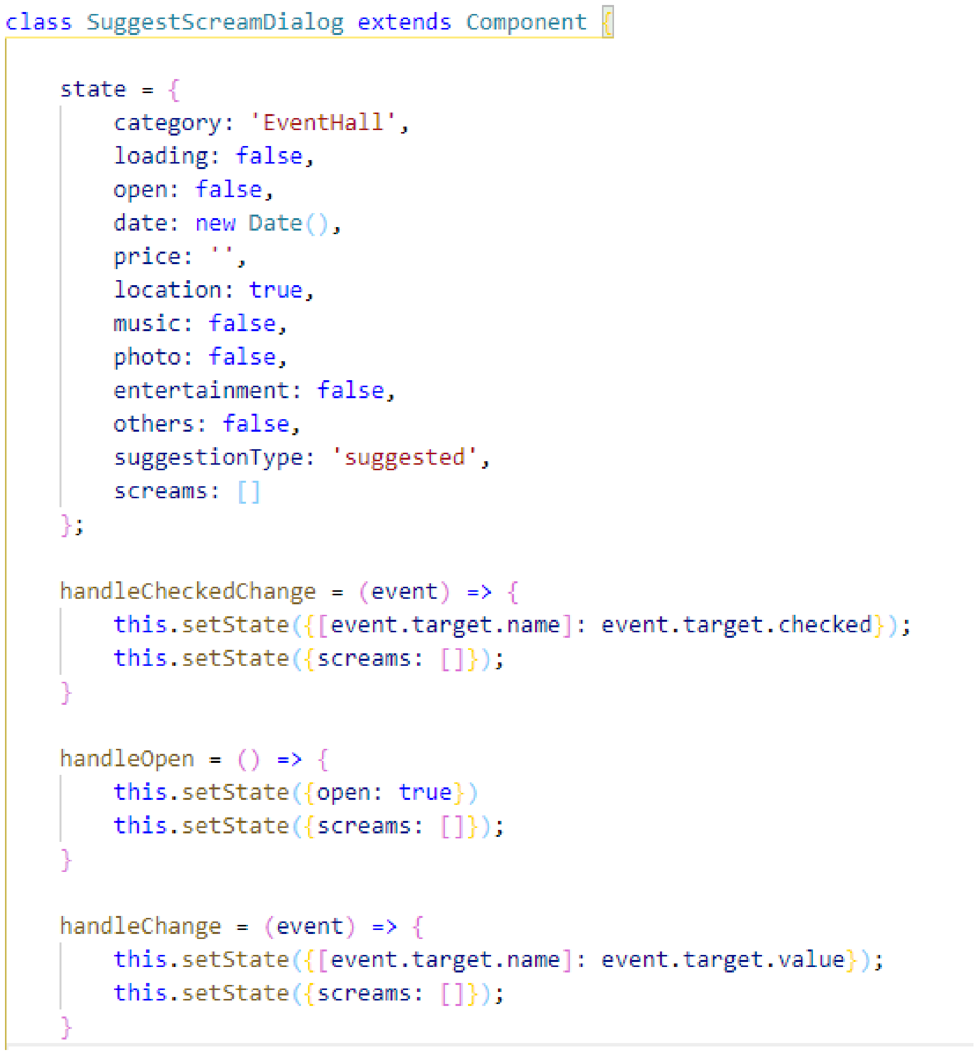


Figure 5.3.4. State Pattern

State is a behavioral design pattern that lets an object alter its behavior when its internal state changes. It appears as if the object changed its class.



```
class SuggestScreamDialog extends Component {

    state = {
        category: 'EventHall',
        loading: false,
        open: false,
        date: new Date(),
        price: '',
        location: true,
        music: false,
        photo: false,
        entertainment: false,
        others: false,
        suggestionType: 'suggested',
        screams: []
    };

    handleCheckedChange = (event) => {
        this.setState({[event.target.name]: event.target.checked});
        this.setState({screams: []});
    }

    handleOpen = () => {
        this.setState({open: true})
        this.setState({screams: []});
    }

    handleChange = (event) => {
        this.setState({[event.target.name]: event.target.value});
        this.setState({screams: []});
    }
```

Figure 5.3.5. State Pattern used in most components

6. Diving into the Wedding Booking website

Wedding
Bookings

Figure 6.1. Logo

On this platform, two types of users can be signed up providing a valid email, a unique username, password and the user type: Married To Be or Service Provider. In the login page the user needs to provide the email and the password of the account that has been created. If login or signing up was successfully done we can dive to the main page, the home page.

6.1. User Manual

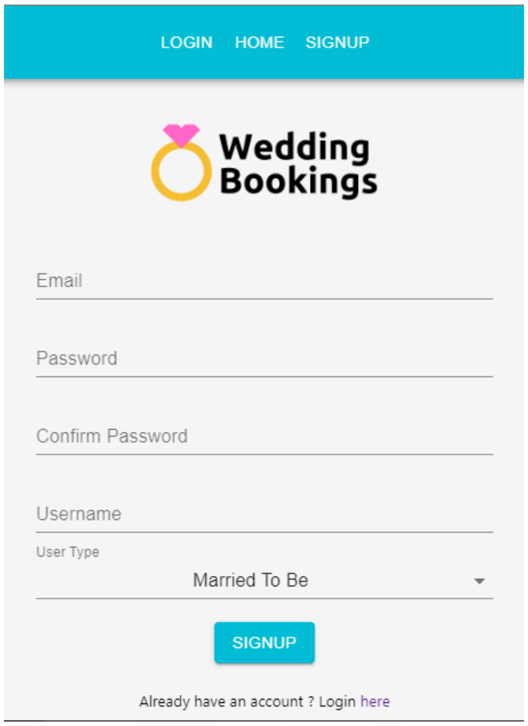


Figure 6.1.1. Signup

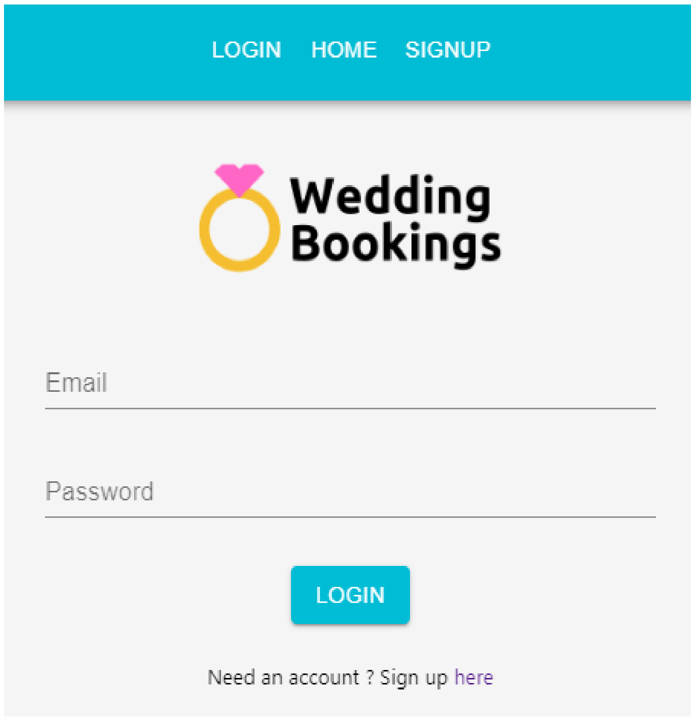


Figure 6.1.2. Login

There are two types of users: **Service Provider** and **Married to Be**. Each user type can do several actions on the website.

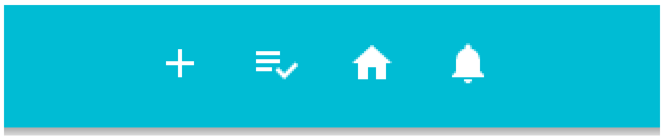
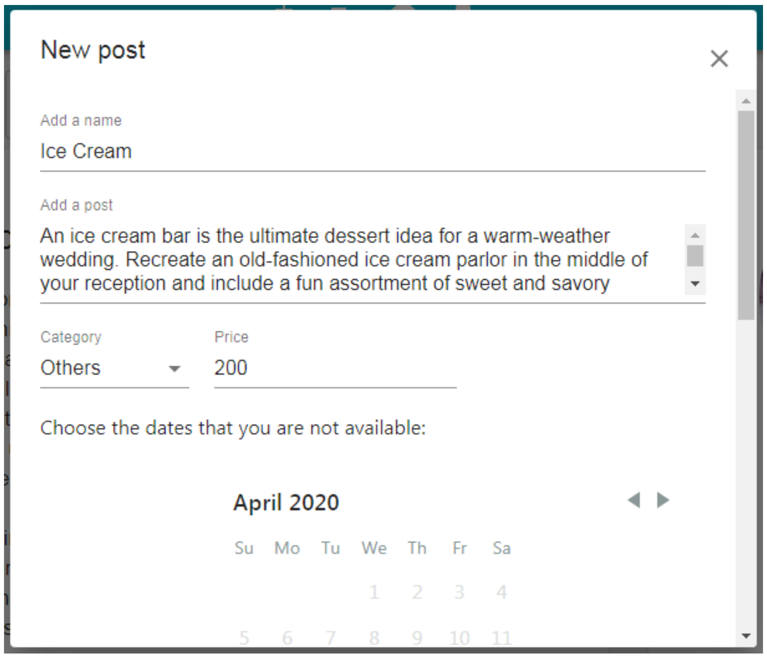


Figure 6.1.3. Navigation Menu

+ New Post! This option will be available only to Service Providers. Here the users can create a new post where they can describe the service to be added.


New post

Add a name
Ice Cream

Add a post
An ice cream bar is the ultimate dessert idea for a warm-weather wedding. Recreate an old-fashioned ice cream parlor in the middle of your reception and include a fun assortment of sweet and savory

Category
Others

Price
200

Choose the dates that you are not available:

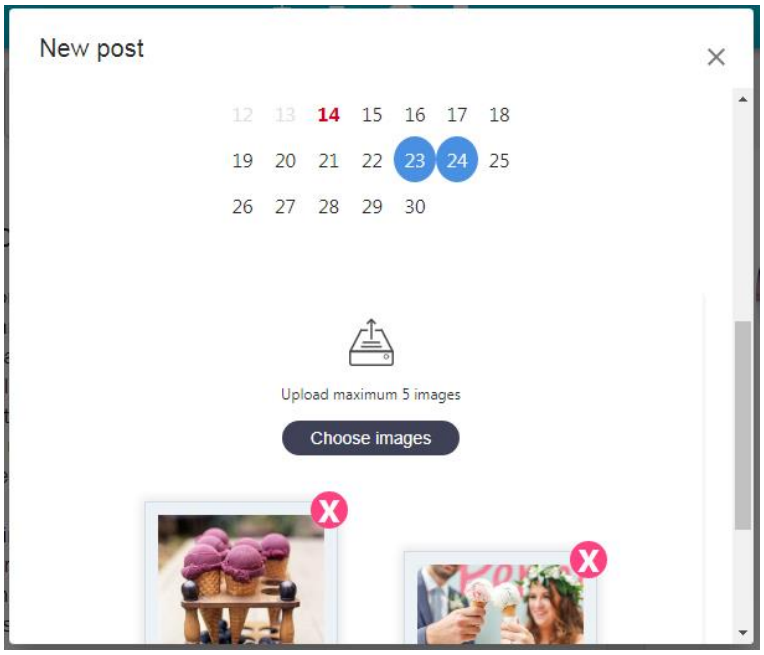


Figure 6.1.4. Adding a new post

Bookings List! This button redirects us to the page where we can see our booked services, in the order of the first service to occur.

Home Page! This is the home page button. There, users will see all the services that are provided by the vendors.

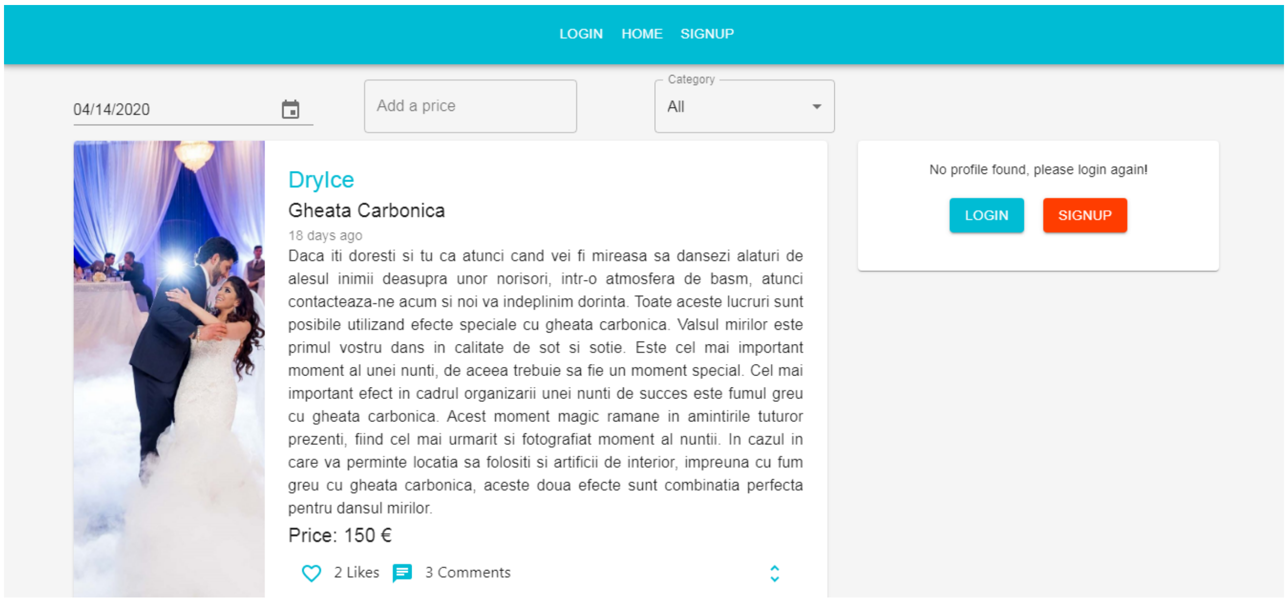
Notifications! Here the users will receive notifications if someone liked, commented or booked a post.

Suggestions! The most important thing in organizing a wedding, is the vendor and service choosing. Get the best vendors from our suggestion algorithm.

On the upper part of the home page we can see 3 types of filtering. Those criterias have been prove to be the most significant parts that a couple lookup for the first time for the wedding.

It’s about :

1. The DATE;
2. The Maximum Price for the service;
3. The category of the service that they are looking for.



04/14/2020

Category: All

DryIce

Gheata Carbonica

18 days ago

Daca iti doresti si tu ca atunci cand vei fi mireasa sa dansezi alaturi de alesul inimii deasupra unor norisori, intr-o atmosfera de basm, atunci contacteaza-ne acum si noi va indeplinim dorinta. Toate aceste lucruri sunt posibile utilizand efecte speciale cu gheata carbonica. Valsul mirilor este primul vostru dans in calitate de sot si sotie. Este cel mai important moment al unei nunti, de aceea trebuie sa fie un moment special. Cel mai important efect in cadrul organizarii unei nunti de succes este fumul greu cu gheata carbonica. Acest moment magic ramane in amintirile tuturor prezenti, fiind cel mai urmarit si fotografiat moment al nuntii. In cazul in care va perminte locatia sa folositi si artificii de interior, impreuna cu fum greu cu gheata carbonica, aceste doua efecte sunt combinatia perfecta pentru dansul mirilor.

Price: 150 €

2 Likes 3 Comments

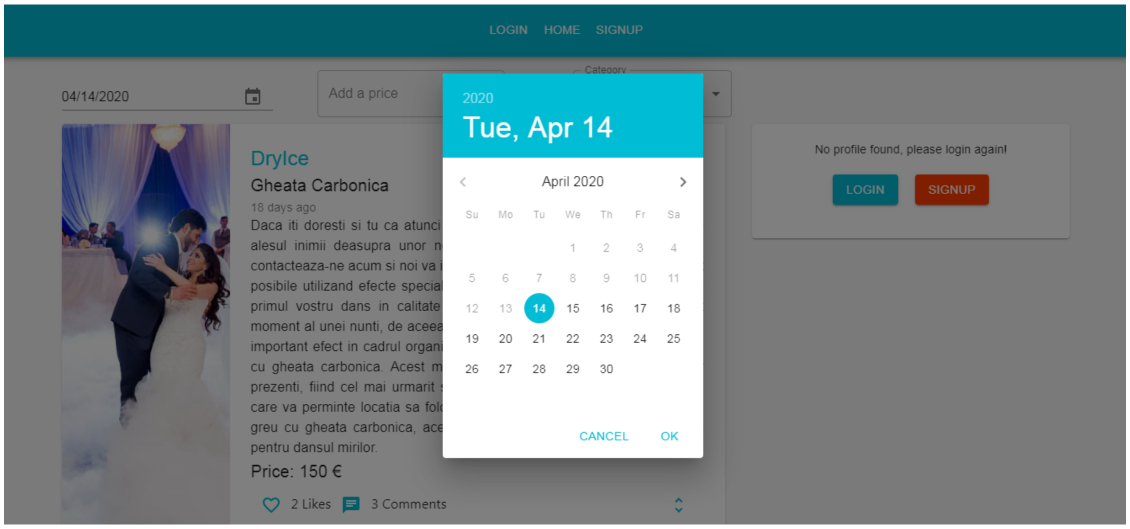
No profile found, please login again!

Figure 6.1.5. Home Page

6.1.1. The MOMENT

Figure 6.1.1.1. The Special Date

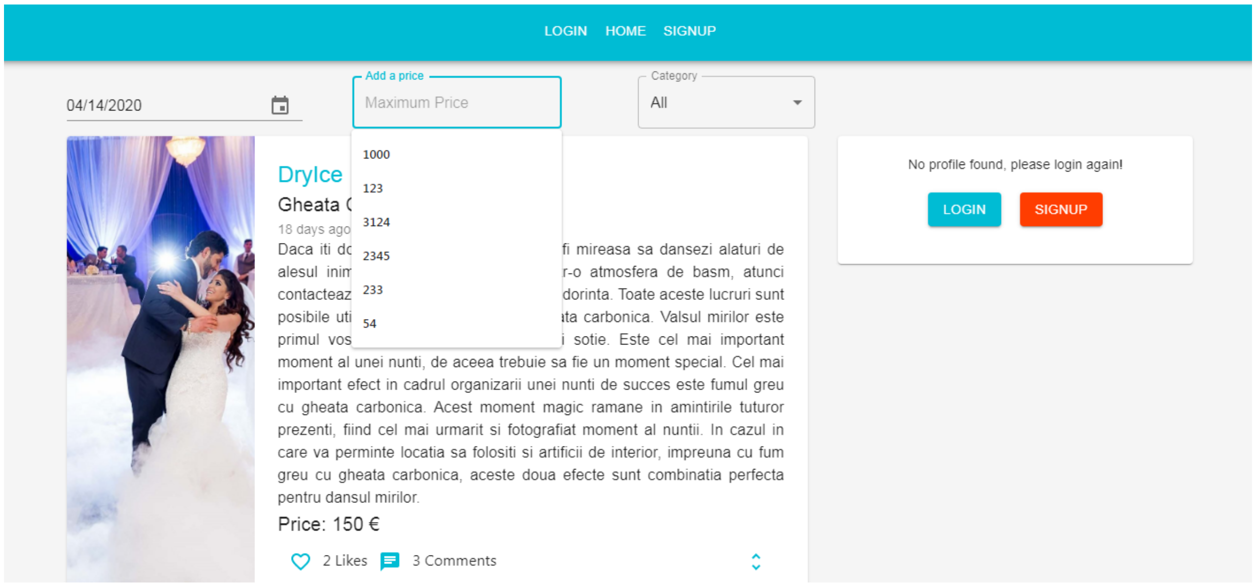
Choosing a wedding date can be a little bit like putting together a puzzle. Consider whether there are any specific dates that are special to you and your significant other. Many wedding dates are related to a special day, like a birthday, the first day that you met, your first date to the movies, or the first time you kissed.



DryIce
Gheata Carbonica
18 days ago
Daca iti doresti si tu ca atunci
alesul inimii deasupra unor n
contacteaza-ne acum si noi va
posibile utilizand efecte special
primul vostru dans in calitate
moment al unei nunti, de aceea
important efect in cadrul organi
cu gheata carbonica. Acest m
prezenti, fiind cel mai urmarit s
care va permite locatia sa fol
greu cu gheata carbonica, ace
pentru dansul mirilor.
Price: 150 €

Figure 6.1.1.2. Choosing Date

6.1.2. The EXPANSES



DryIce
Gheata C
18 days ago
Daca iti do ... fi mireasa sa dansezi alaturi de
alesul inim ... r-o atmosfera de basm, atunci
contacteaz ... dorinta. Toate aceste lucruri sunt
posibile uti ... ita carbonica. Valsul mirilor este
primul vos ... i sotie. Este cel mai important
moment al unei nunti, de aceea trebuie sa fie un moment special. Cel mai
important efect in cadrul organizarii unei nunti de succes este fumul greu
cu gheata carbonica. Acest moment magic ramane in amintirile tuturor
prezenti, fiind cel mai urmarit si fotografiat moment al nuntii. In cazul in
care va permite locatia sa folositi si artificii de interior, impreuna cu fum
greu cu gheata carbonica, aceste doua efecte sunt combinatia perfecta
pentru dansul mirilor.
Price: 150 €

Figure 6.1.2.1 Maximum Price for the Service

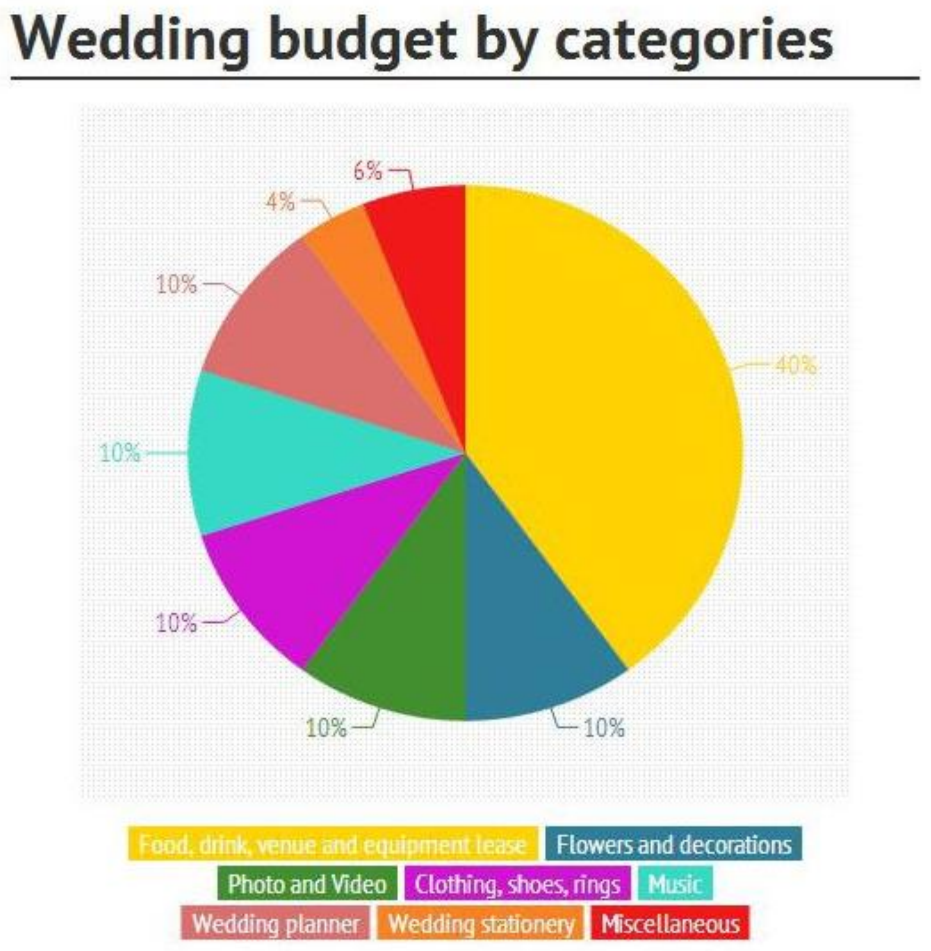


Figure 6.1.2.2. The budget of each service

Figuring out your wedding budget is no easy feat. Your wedding will likely be the biggest party you've ever hosted and the priciest. To make a budget, you'll need to tally up your savings, maintain a detailed spreadsheet so you don't go over during the planning process, prepare for unexpected costs, as well as make meaningful cuts if you do exceed your total budget. It's hard work, but putting in the time and energy now ensures you'll live happily ever after. Here at Wedding Booking you can search for the vendor and the service you really want for the budget that you really have. So, going over the budget will no longer be a problem. (13)

6.1.3. Vendors Categories

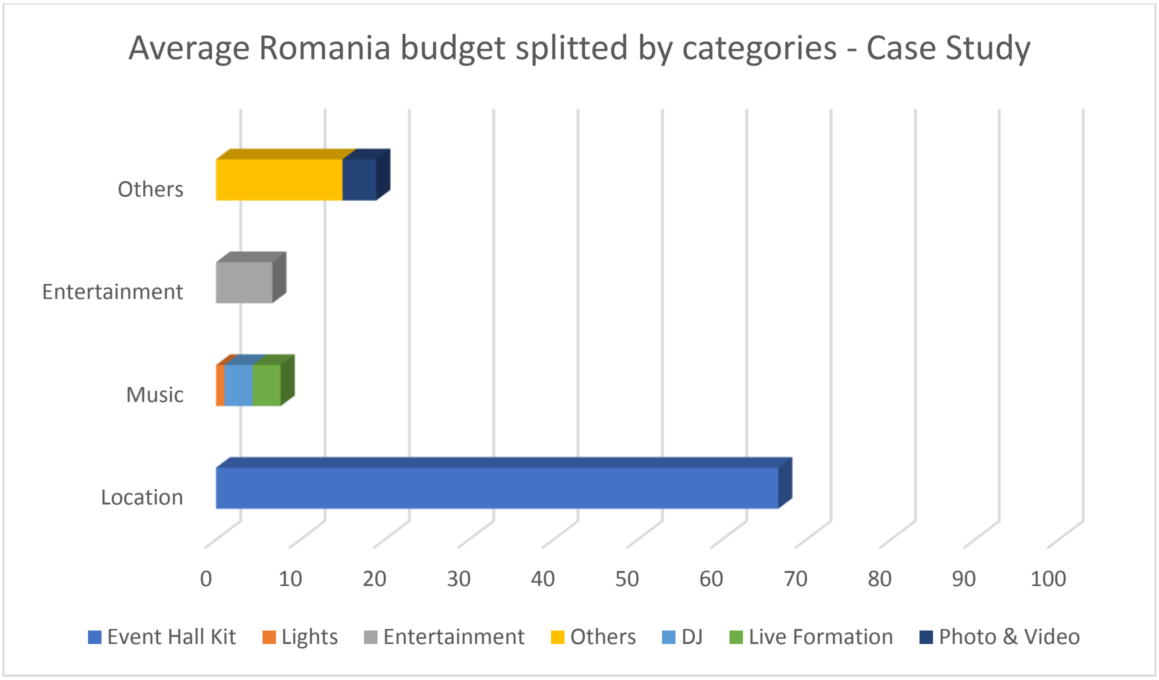
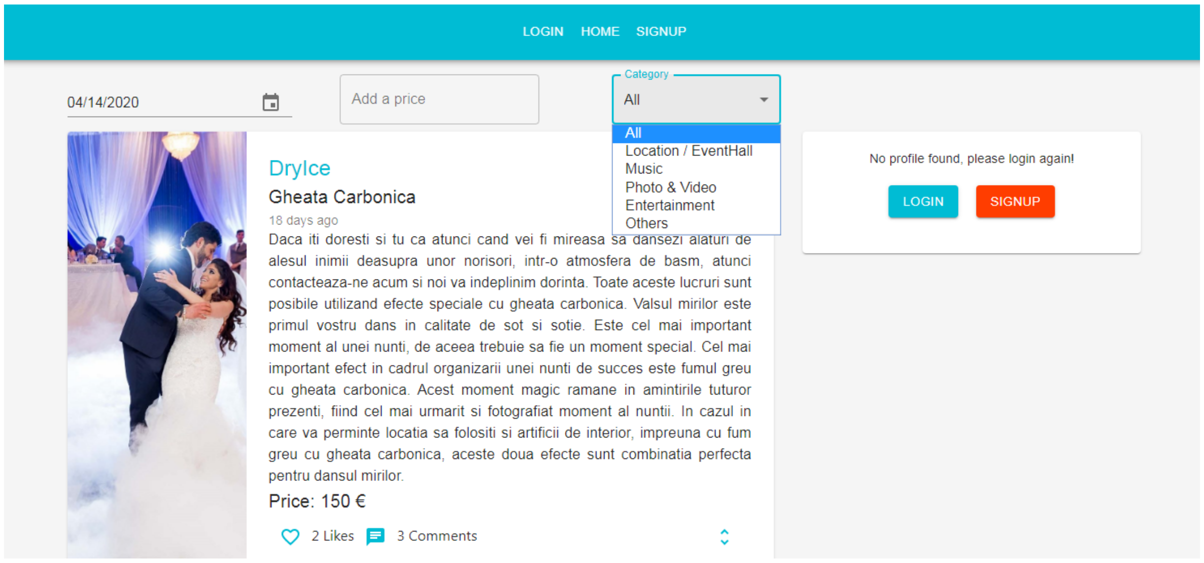


Figure 6.1.3.1. Personal Case Study

Perhaps the most important part of wedding planning is building your vendor team, hiring the group of professionals who will make your vision a reality. If you're just getting started, you're probably asking yourself the big question: Who should be on your wedding vendor list? What types of vendors do I actually need to make my big day a success? According to the 2019 WeddingWire Newlywed Report, couples hire 14 vendors for their wedding day, on average. (14)


LOGIN HOME SIGNUP

04/14/2020

Add a price

Category

All

All
Location / EventHall
Music
Photo & Video
Entertainment
Others

DryIce

Gheata Carbonica

18 days ago

Daca iti doresti si tu ca atunci cand vei fi mireasa sa dansezi alaturi de alesul inimii deasupra unor norisori, intr-o atmosfera de basm, atunci contacteaza-ne acum si noi va indeplinim dorinta. Toate aceste lucruri sunt posibile utilizand efecte speciale cu gheata carbonica. Valsul mirilor este primul vostru dans in calitate de sot si sotie. Este cel mai important moment al unei nunti, de aceea trebuie sa fie un moment special. Cel mai important efect in cadrul organizarii unei nunti de succes este fumul greu cu gheata carbonica. Acest moment magic ramane in amintirile tuturor prezenti, fiind cel mai urmarit si fotografiat moment al nuntii. In cazul in care va permite locatia sa folositi si artificii de interior, impreuna cu fum greu cu gheata carbonica, aceste doua efecte sunt combinatia perfecta pentru dansul mirilor.

Price: 150 €

2 Likes 3 Comments

No profile found, please login again!

LOGIN SIGNUP

Figure 6.1.3.2. Categories

Here we have a list of the most important vendors that you can find on our website:

- Location Vendors,
- DJ’s and Live formations,
- Photo & Video,
- Wedding Planners,
- Cake Bakers,
- Florists,
- Rentals Companies,
- And many more!

But which one is the number one priority? Date, Budget or Location? You've got to find a date that fits in with your schedules and the schedules of your loved ones, while also keeping season, availability, and budget in mind. It's like the stars have to perfectly align in order to settle on a specific wedding date because trying to find a day where your desired venues and vendors are all available is quite a feat in itself. We have the solution. You can check for a specific day, for a specific budget and for a specific venue and vendor, so that the perfect wedding can happen exactly how the groom and the bride wants.

Further down the page we can find two parts:

- In the right hand side there are the logged in user profile details as well as the profile picture;
- In the left side there is the list of posts of the service provider users;

6.1.4. Profile

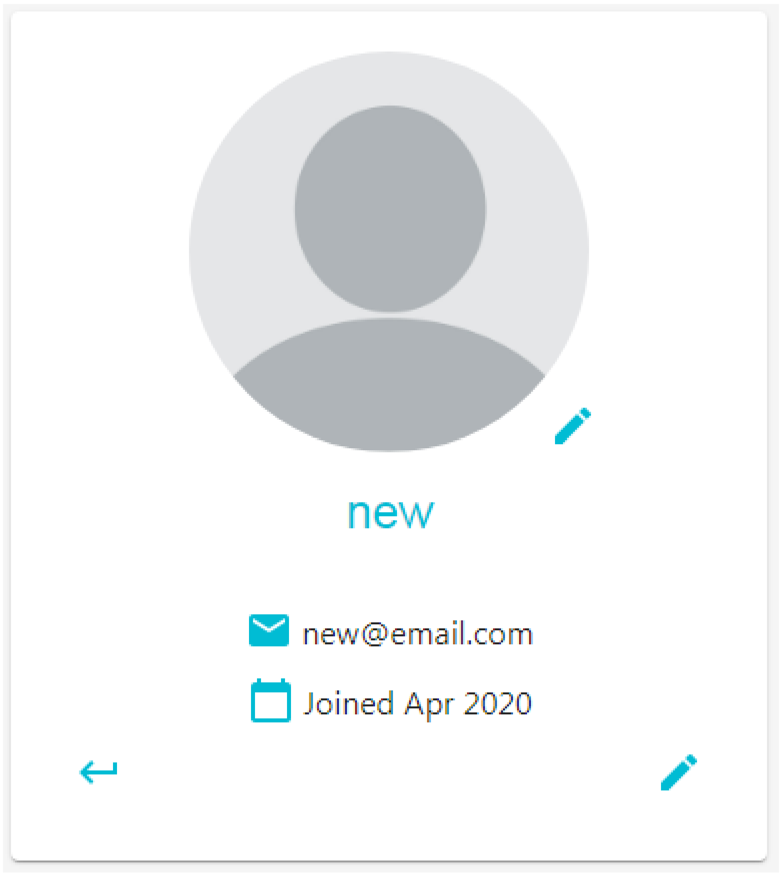
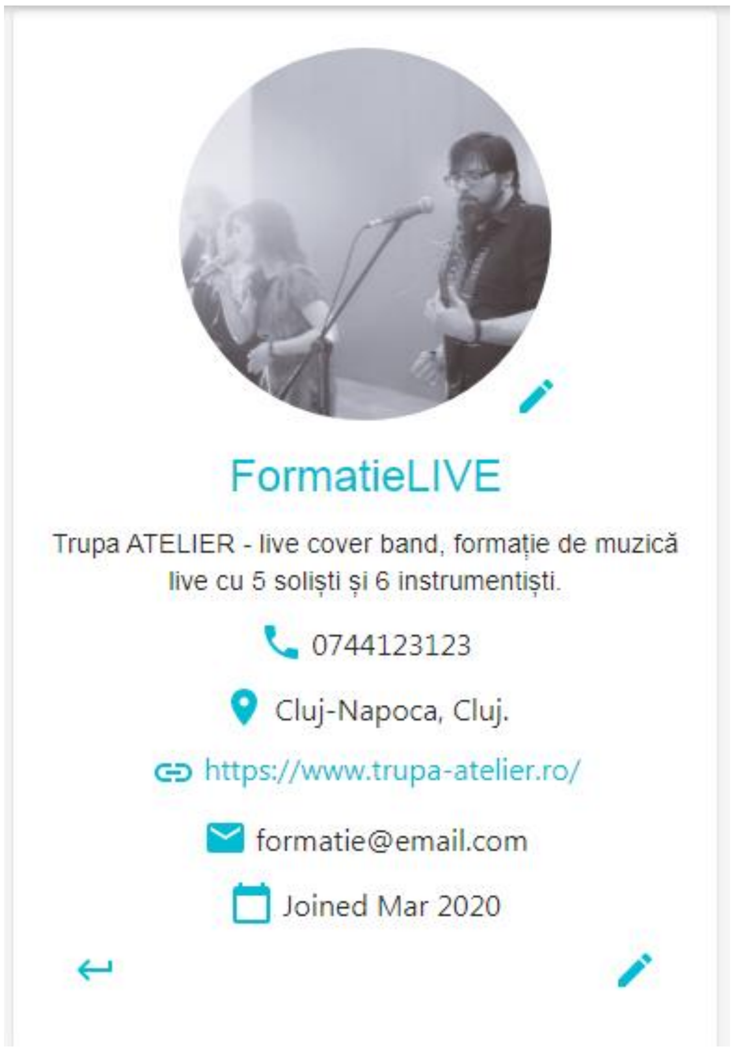


Figure 6.1.4.1. Fresh Account

A newly signed up user will see in the profile box at start a blank avatar image, which is attributed at all new accounts. In order for user to change his/her profile picture we added a button where they can upload a new photo.

Lower down we have the profile details such as:

1. At first:
   1.1. Email;
   1.2. Date of the joined user.
2. Later to be added:
   2.1. Description;
   2.2. Website;
   2.3. Location;
   2.4. Phone.



FormatieLIVE

Trupa ATELIER - live cover band, formație de muzică live cu 5 soliști și 6 instrumentiști.

0744123123

Cluj-Napoca, Cluj.

https://www.trupa-atelier.ro/

formatie@email.com

Joined Mar 2020

Figure 6.1.4.2. Complete Account

To add those profile details, lowermost the profile box, we can find a button that opens a new dialog for user to add those details along with logout button. After the logout button is pressed, there appear two buttons, a login and signup that redirect the user to the respective pages.

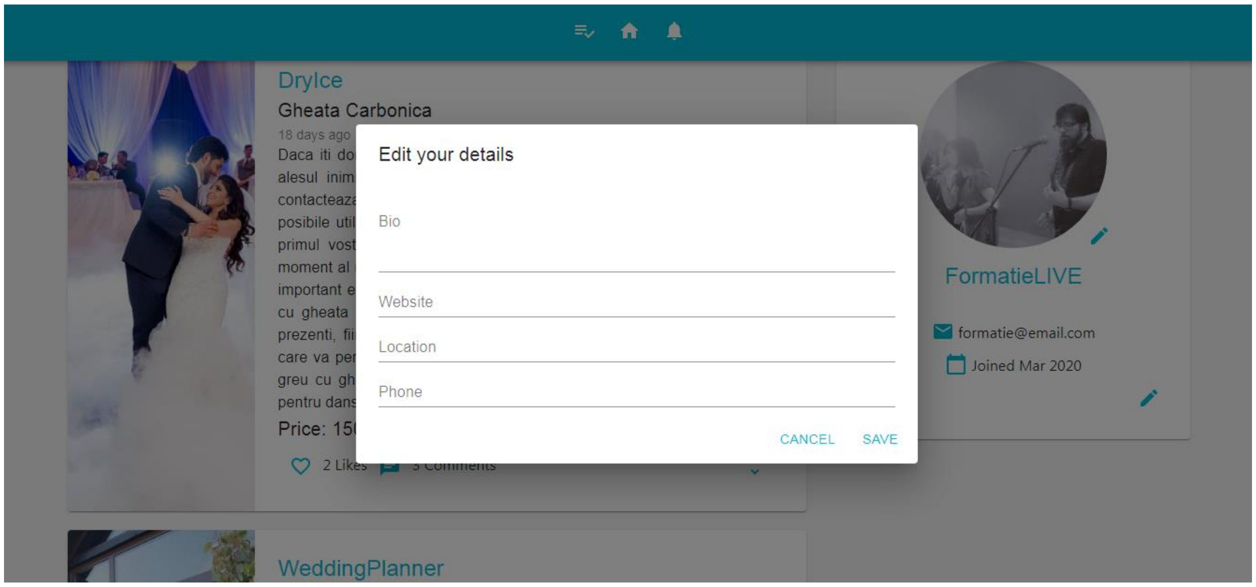


Figure 6.1.4.3. Dialog Add/Edit details

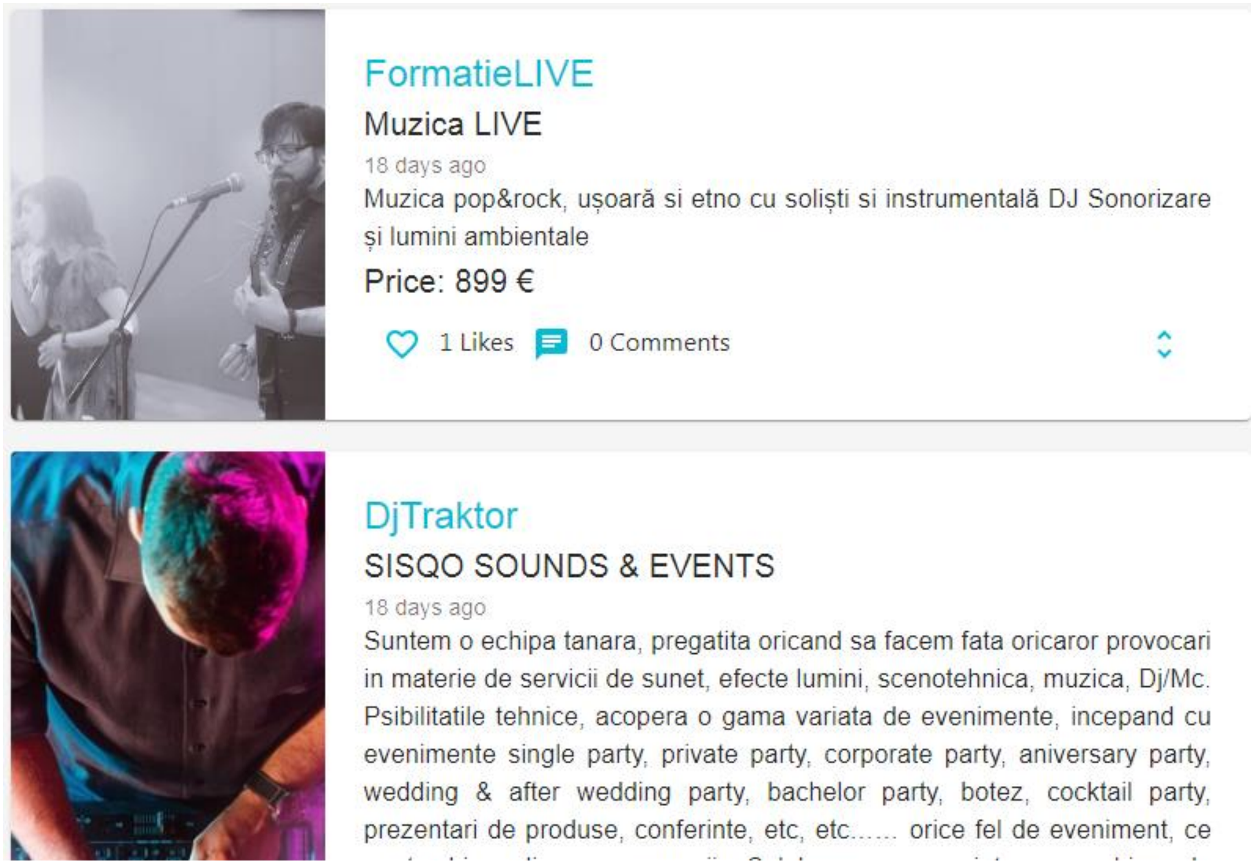
6.1.5. Posts

The posts part contains multiple posts from all the service provider users that have posted a service. One post is divided in to two main subparts: the profile picture of the user that posted the service and the details about the post.

On the details section we can find from top to bottom:

- The user name
- The title of the post

- How long ago the service has been posted
- Description of the service
- Price
- Likes & Comments


FormatieLIVE

Muzica LIVE

18 days ago

Muzica pop&rock, ușoară si etno cu soliști si instrumentală DJ Sonorizare și lumini ambientale

Price: 899 €

DjTraktor

SISQO SOUNDS & EVENTS

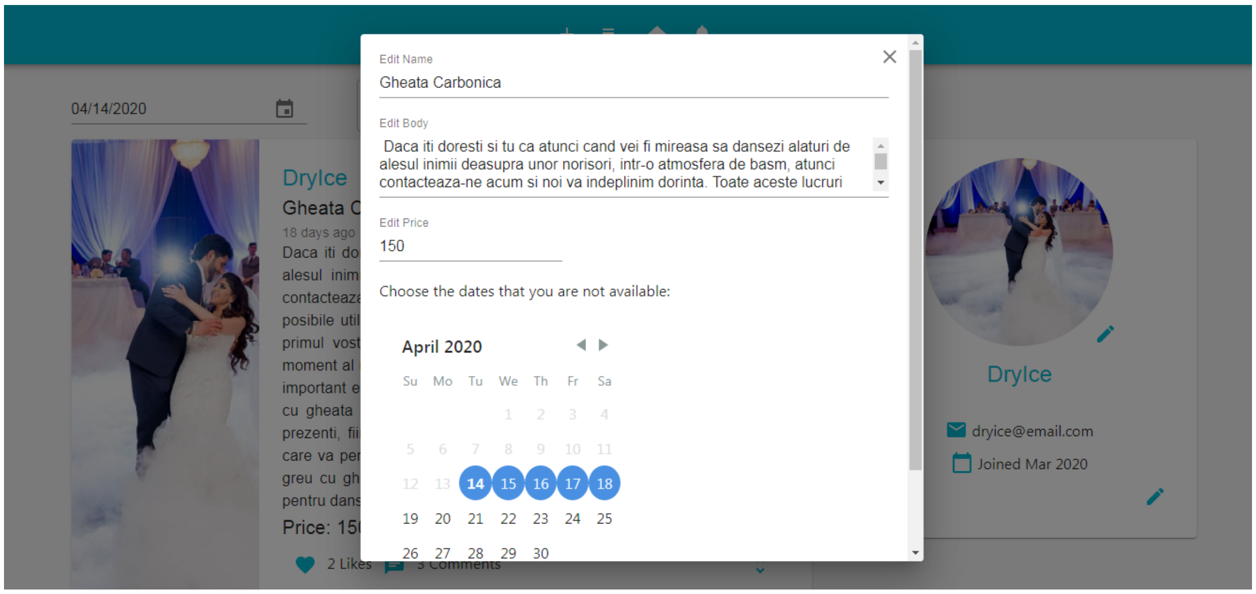
18 days ago

Suntem o echipa tanara, pregatita oricand sa facem fata oricaror provocari in materie de servicii de sunet, efecte lumini, scenotehnica, muzica, Dj/Mc. Psibilitatile tehnice, acopera o gama variata de evenimente, incepand cu evenimente single party, private party, corporate party, aniversary party, wedding & after wedding party, bachelor party, botez, cocktail party, prezentari de produse, conferinte, etc, etc...... orice fel de eveniment, ce

Figure 6.1.5.1. Posts

The owner of the post is able to edit or delete it. Two buttons will be prompt to the posts that were created by the user logged in.

✎ If this icon is clicked a dialog will be opened where the user can edit the name, body, price or the busy dates of the post.


Edit Name

Gheata Carbonica

Edit Body

Daca iti doresti si tu ca atunci cand vei fi mireasa sa dansezi alaturi de alesul inimii deasupra unor norisori, intr-o atmosfera de basm, atunci contacteaza-ne acum si noi va indeplinim dorinta. Toate aceste lucruri

Edit Price

150

Choose the dates that you are not available:

Figure 6.1.5.2. Edit Post

If the user decides to delete the post, he can click on the icon and confirm it in the next dialog

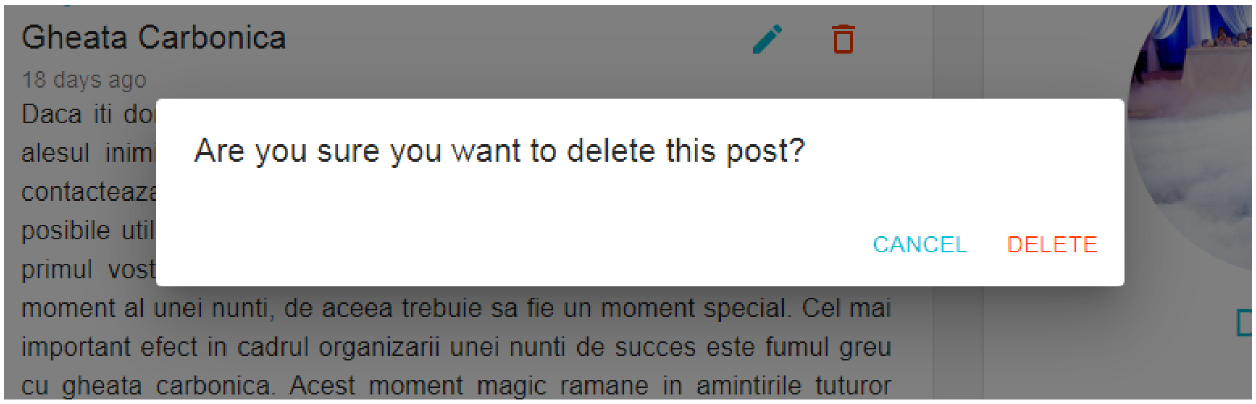
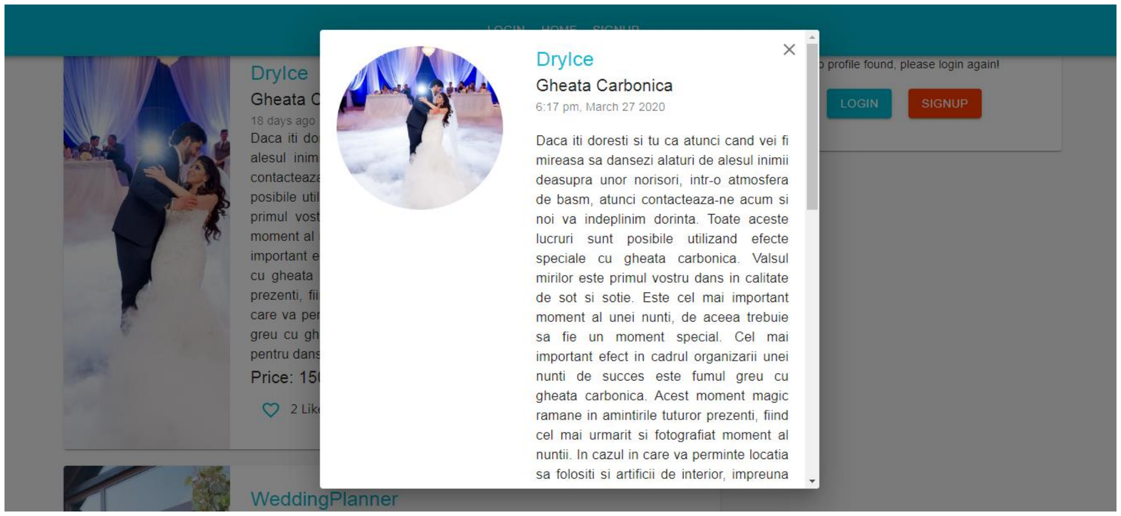

Are you sure you want to delete this post?

Figure 6.1.5.3. Delete Post

Other users can like the posts by clicking on the empty heart transforming it to a filled heart. There is an "Extend Post" button where users can see other details related to the post, such as: images uploaded by the vendors and a comment form where users can comment and see comments from other users.



DryIce

Gheata Carbonica

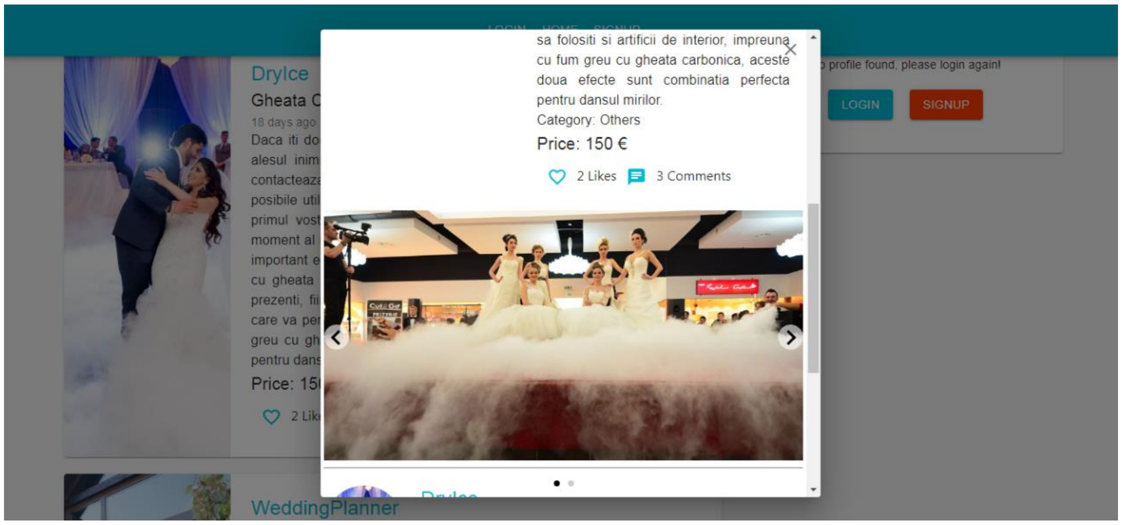
6:17 pm, March 27 2020

Daca iti doresti si tu ca atunci cand vei fi mireasa sa dansezi alaturi de alesul inimii deasupra unor norisori, intr-o atmosfera de basm, atunci contacteaza-ne acum si noi va indeplinim dorinta. Toate aceste lucruri sunt posibile utilizand efecte speciale cu gheata carbonica. Valsul mirilor este primul vostru dans in calitate de sot si sotie. Este cel mai important moment al unei nunti, de aceea trebuie sa fie un moment special. Cel mai important efect in cadrul organizarii unei nunti de succes este fumul greu cu gheata carbonica. Acest moment magic ramane in amintirile tuturor prezenti, fiind cel mai urmarit si fotografiat moment al nuntii. In cazul in care va permite locatia sa folositi si artificii de interior, impreuna

o profile found, please login again!

LOGIN SIGNUP

WeddingPlanner

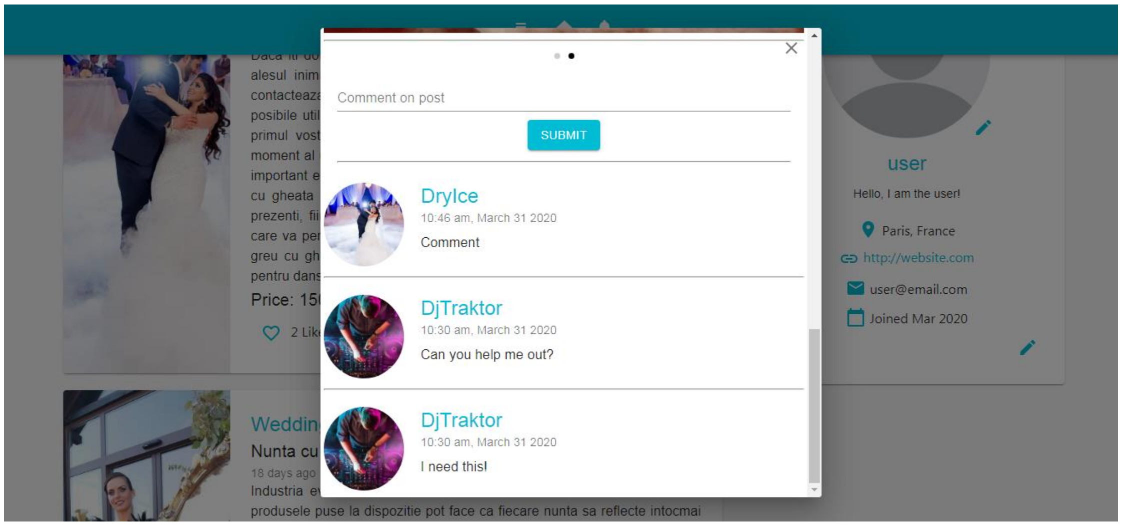


sa folositi si artificii de interior, impreuna cu fum greu cu gheata carbonica, aceste doua efecte sunt combinatia perfecta pentru dansul mirilor.

Category: Others

Price: 150 €

2 Likes 3 Comments



Comment on post

SUBMIT

DryIce
10:46 am, March 31 2020
Comment

DjTraktor
10:30 am, March 31 2020
Can you help me out?

DjTraktor
10:30 am, March 31 2020
I need this!

user

Hello, I am the user!

Paris, France

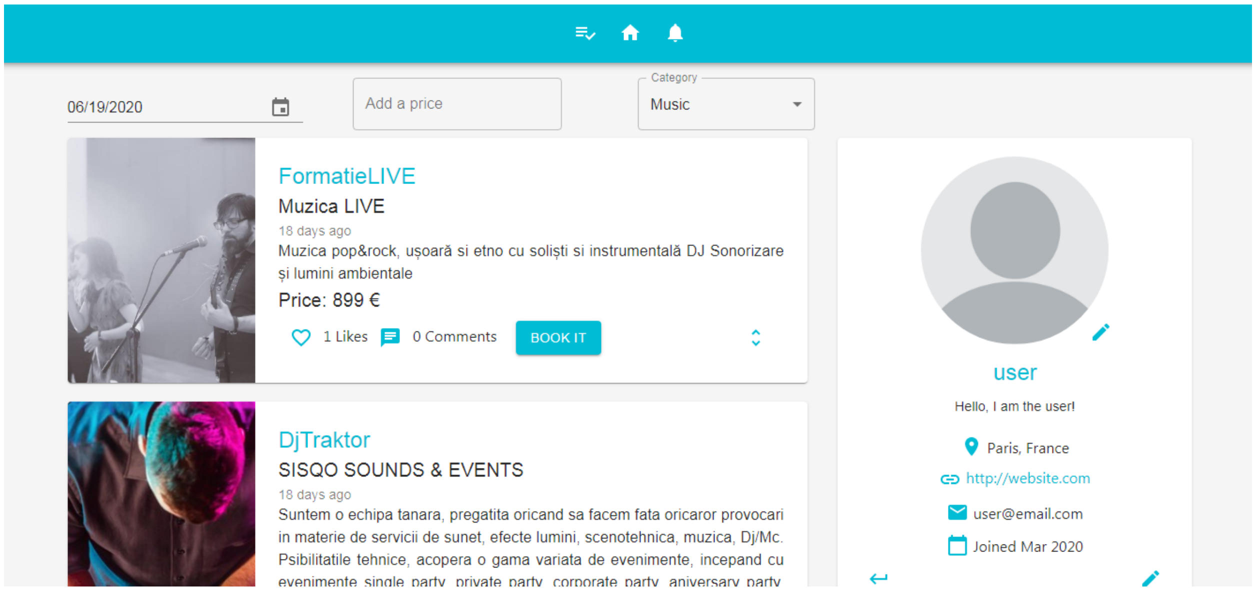
http://website.com

user@email.com

Joined Mar 2020

Figure 6.1.5.4. Expand Post & Comment Section

The users that are married to be, can book services from the home page by choosing the date and other filter actions. After that, they can watch the list of bookings from the navigation menu.


06/19/2020

Add a price

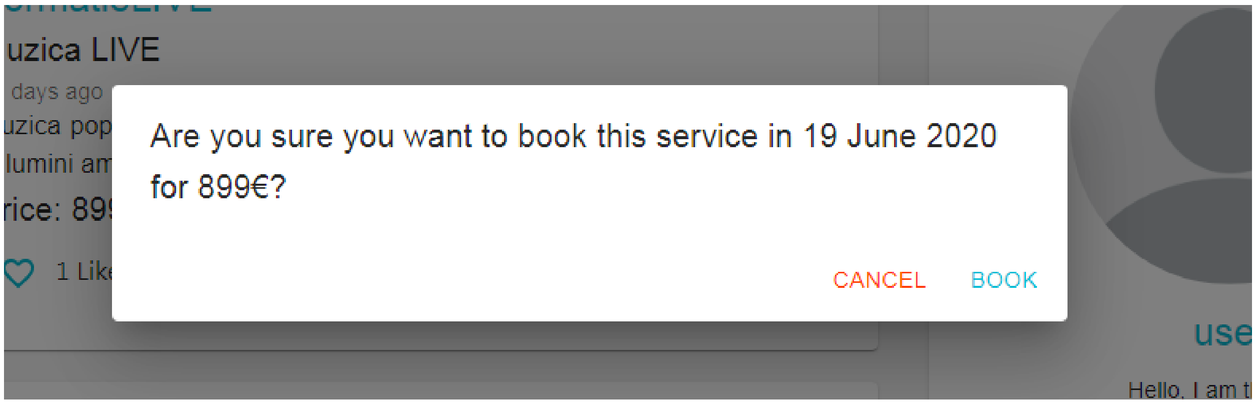
Category
Music

FormatieLIVE
Muzica LIVE
18 days ago
Muzica pop&rock, ușoară si etno cu soliști si instrumentală DJ Sonorizare și lumini ambientale
Price: 899 €
1 Likes 0 Comments BOOK IT

DjTraktor
SISQO SOUNDS & EVENTS
18 days ago
Suntem o echipa tanara, pregatita oricand sa facem fata oricaror provocari in materie de servicii de sunet, efecte lumini, scenotehnica, muzica, Dj/Mc. Psibilitatile tehnice, acopera o gama variata de evenimente, incepand cu evenimente single party, private party, corporate party, aniversary party

user
Hello, I am the user!
Paris, France
http://website.com
user@email.com
Joined Mar 2020

Figure 6.1.5.5. BOOK IT button


uzica LIVE
days ago
uzica pop
lumini am
rice: 89
1 Like

Are you sure you want to book this service in 19 June 2020 for 899€?

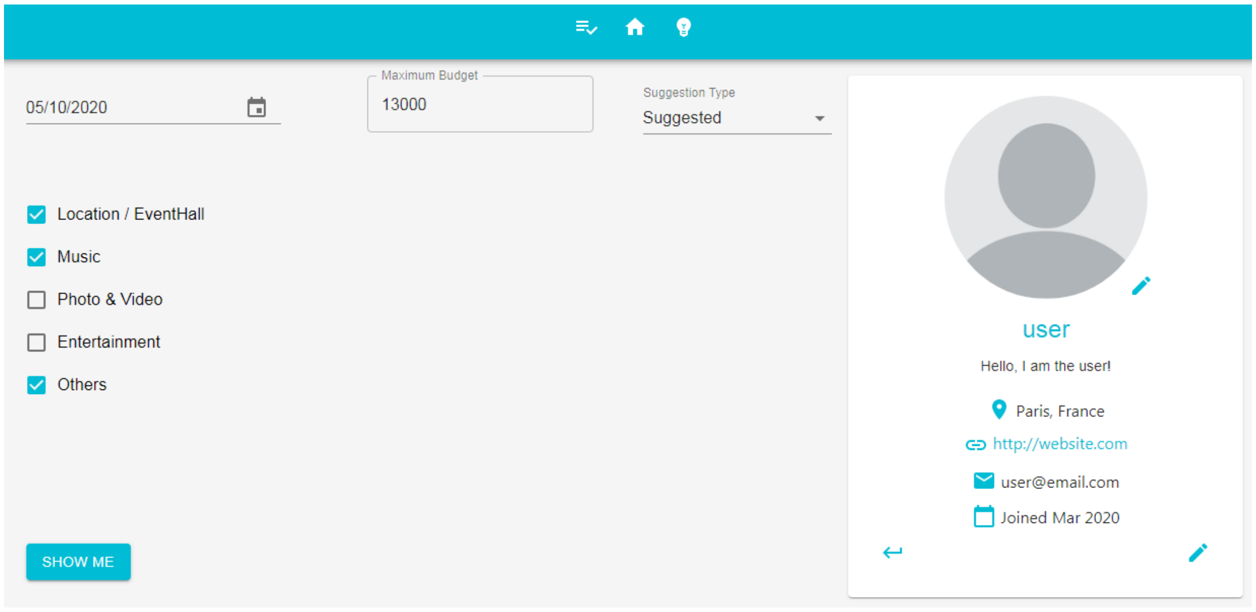
CANCEL BOOK

use
Hello, I am t

Figure 6.1.5.6. Book Confirmation

6.1.6. Suggest Package

On the **Suggest Package** only ‘Married to be’ users can join. Here they will find the most important searching criterias for a wedding day.


05/10/2020

Maximum Budget: 13000

Suggestion Type: Suggested

- [x] Location / EventHall
- [x] Music
- [ ] Photo & Video
- [ ] Entertainment
- [x] Others

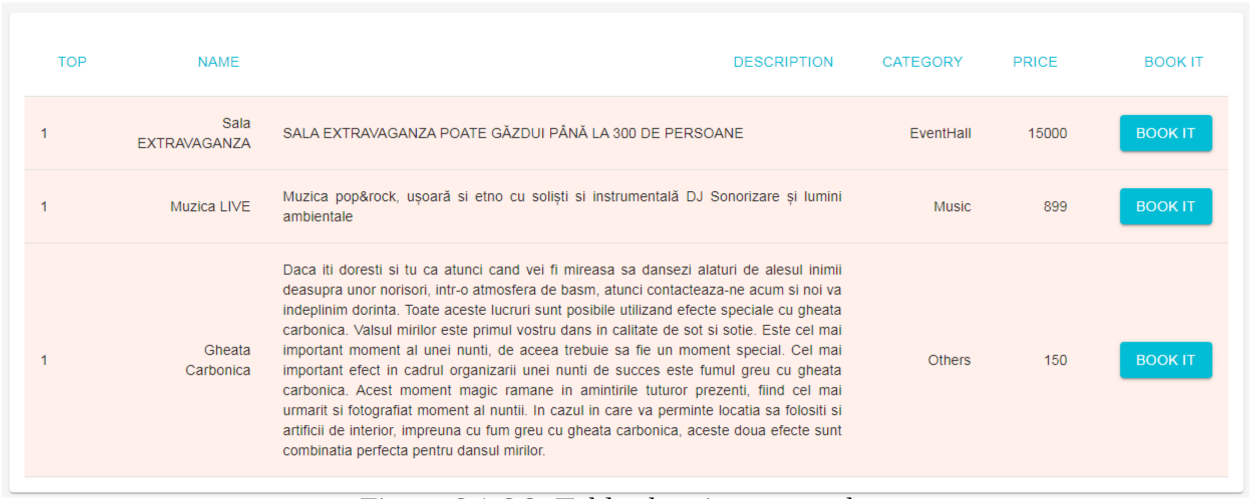
user

Hello, I am the user!

Paris, France
http://website.com
user@email.com
Joined Mar 2020

Figure 6.1.6.1. Criterias

After the criterias have been chosen by the users, they are a click away from showing some services and creating a wedding package. After the packages are shown they can choose to book what services are they interested in from the top two selected packages.



TOP	NAME	DESCRIPTION	CATEGORY	PRICE	BOOK IT
1	Sala EXTRAVAGANZA	SALA EXTRAVAGANZA POATE GĂZDUI PÂNĂ LA 300 DE PERSOANE	EventHall	15000	BOOK IT
1	Muzica LIVE	Muzica pop&rock, ușoară si etno cu soliști si instrumentală DJ Sonorizare și lumini ambientale	Music	899	BOOK IT
1	Gheata Carbonica	Daca iti doresti si tu ca atunci cand vei fi mireasa sa dansezi alaturi de alesul inimii deasupra unor norisori, intr-o atmosfera de basm, atunci contacteaza-ne acum si noi va indeplinim dorinta. Toate aceste lucruri sunt posibile utilizand efecte speciale cu gheata carbonica. Valsul mirilor este primul vostru dans in calitate de sot si sotie. Este cel mai important moment al unei nunti, de aceea trebuie sa fie un moment special. Cel mai important efect in cadrul organizarii unei nunti de succes este fumul greu cu gheata carbonica. Acest moment magic ramane in amintirile tuturor prezenti, fiind cel mai urmarit si fotografiat moment al nuntii. In cazul in care va perminte locatia sa folositi si artificii de interior, impreuna cu fum greu cu gheata carbonica, aceste doua efecte sunt combinatia perfecta pentru dansul mirilor.	Others	150	BOOK IT

Figure 6.1.6.2. Table showing top packages

7.Diagrams

7.1. Use Case Diagram

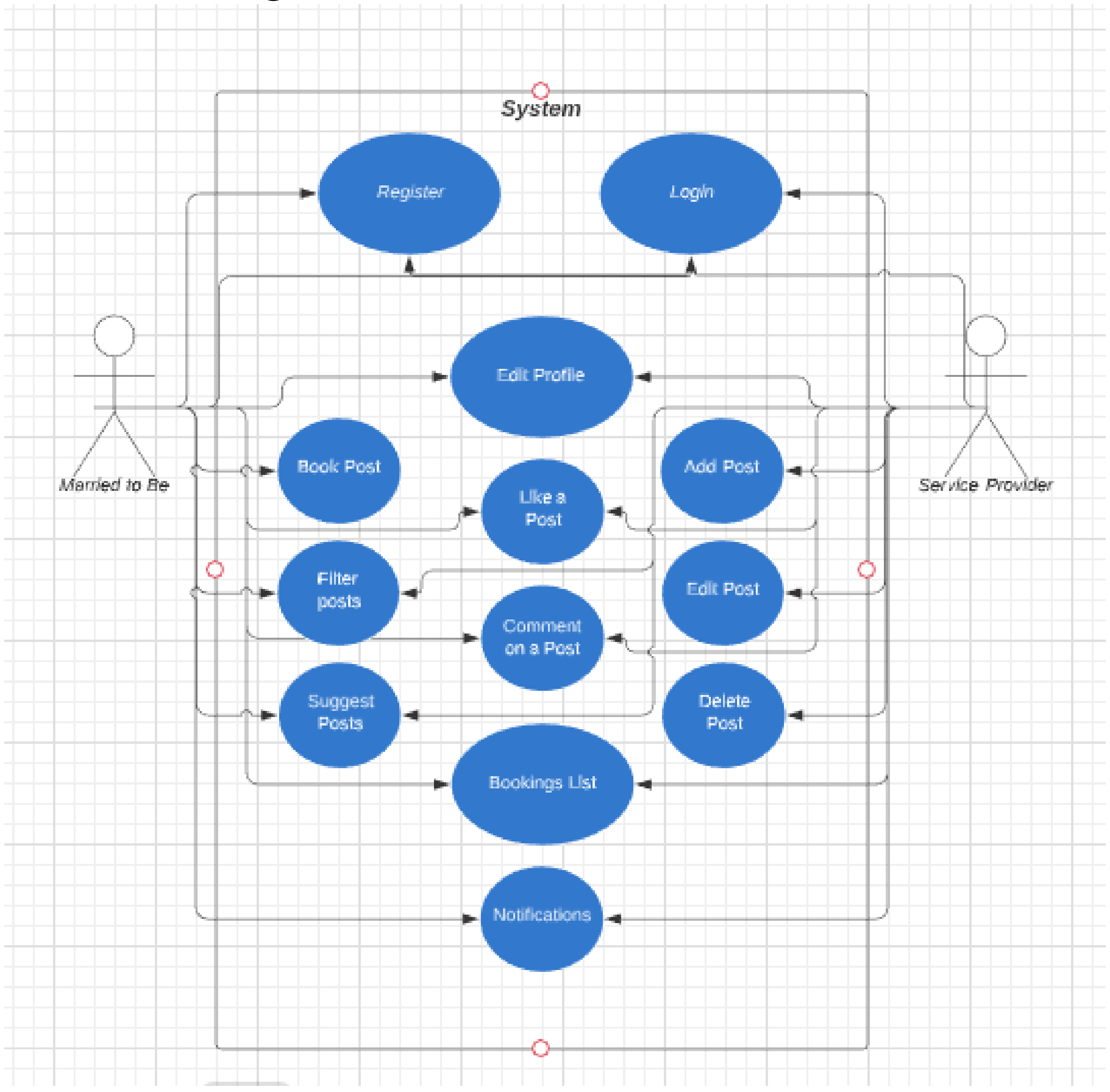


Figure 7.1. Use Case Diagram

7.2. Sequence Diagram

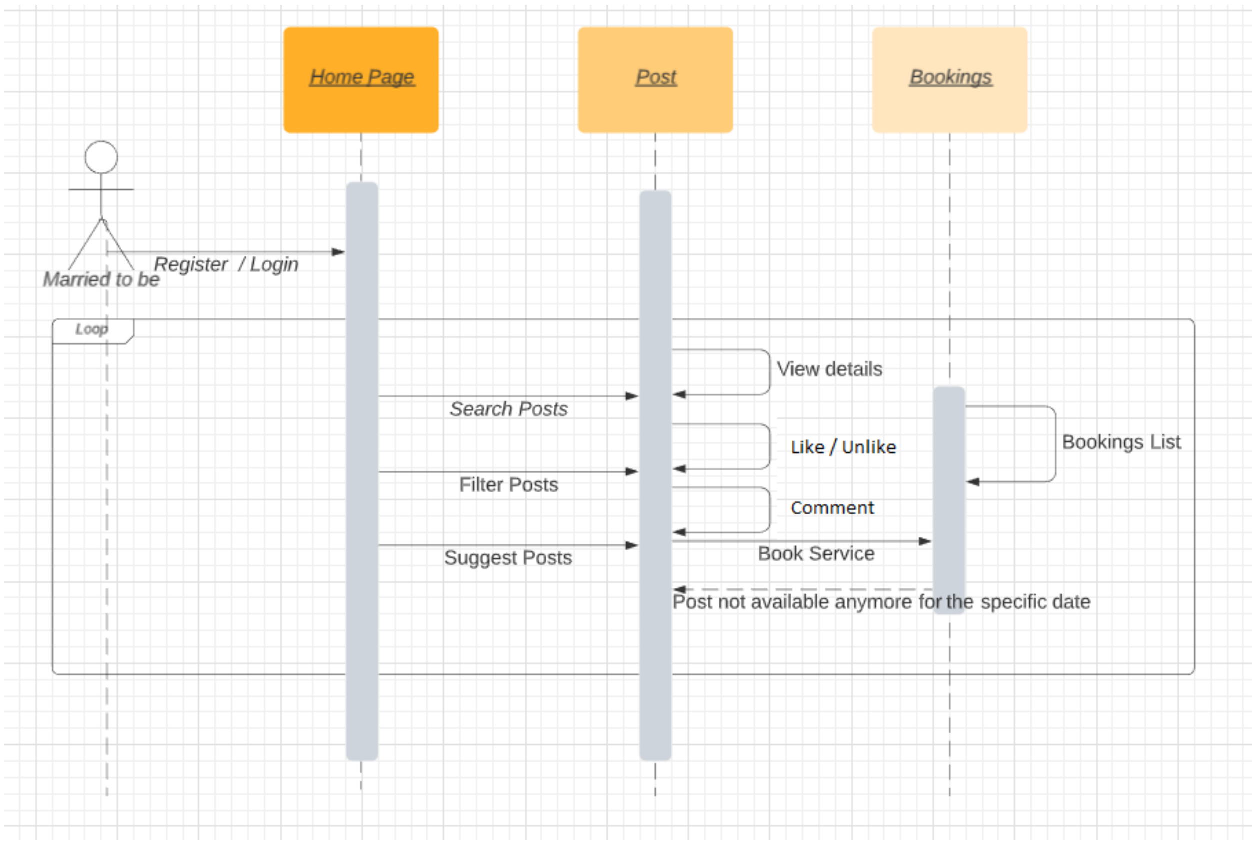


Figure 7.2.1. Sequence diagram - Married to be

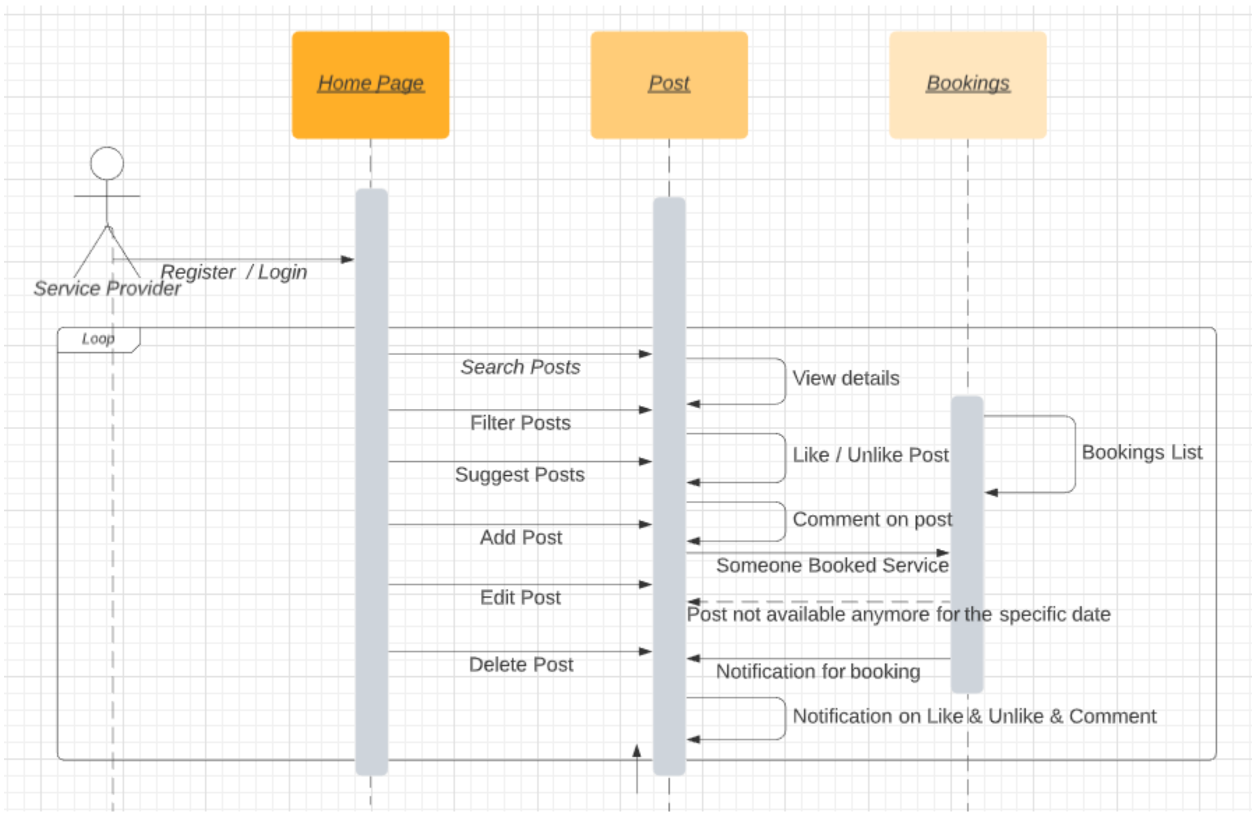


Figure 7.2.2. Sequence diagram – Service Provider

8. Conclusion

Wedding Booking is an easy-to-use website with an extraordinary collection of event halls, vendors and more, for managing wedding planning tasks and creating the best wedding day everyone deserves. Explore the latest trendings, search for perfect services and book through Wedding Booking.

8.1 What extra value does this website brings?

This project is about promoting happiness and celebrating life through highlighting exceptional spaces, places and more.

- ✓ Search services for specific dates and budgets are one click away;
- ✓ Discover the “best fit” with the suggestion algorithm;
- ✓ Book and manage an entire collection of services within a few easy steps;
- ✓ Engage with the vendors by liking and commenting on services posts;
- ✓ No stress
- ✓ A lot less time consuming rather than searching for vendors all over the city;
- ✓ All offers at one place
- ✓ Determine a budget and stick with it
- ✓ Everything ready in anytime of the day
- ✓ No appointments
- ✓ Easy wedding preparations for all couples
- ✓ Compare offers
- ✓ Perfect wedding at clicks ahead
- ✓ An extra place for the vendors to promote their services

8.2. Future improvements

Firstly, one of the future work that can upgrade this project is improving the suggestion algorithm by creating a good and bad factor for the reviews with Neuro-Linguistic Programming. With those quality comments we can see which service is better on the comments engagement.

Secondly, such a website does not have any boundaries so it should be easy for us to extend it worldwide. Currently we are using Firebase Hosting which is free, so we can add a custom domain (.com).

Lastly, we can create a mobile application regarding our website so that users can surf through our application easier, for both Android and iOS.

Bibliography

1. Project Management: Planning a Wedding. *Education Index.* [Online] https://www.educationindex.com/essay/Project-Management-Planning-a-Wedding-PKJKFS7E4Z.

2. **24, Essay by.** Wedding Project Part II. *Essays24.* [Online] December 19, 2010. https://www.essays24.com/essay/Wedding-Project-Part-Ii/22791.html.

3. Node.js. *Wikipedia.* [Online] https://en.wikipedia.org/wiki/Node.js.

4. **Esplin, Chris**. *Medium.* [Online] Oct 24, 2016. https://howtofirebase.com/what-is-firebase-fcb8614ba442.

5. *ReactJS.* [Online] https://reactjs.org/docs.

6. **Realm, Code**. Meet Material-UI — your new favorite user interface library. [Online] April 28, 2018. https://www.freecodecamp.org/news/meet-your-material-ui-your-new-favorite-user-interface-library-6349a1c88a8c/.

7. **Borrelli, Piero**. *LogRocket.* [Online] Oct 15, 2019. https://blog.logrocket.com/the-perfect-architecture-flow-for-your-next-node-js-project/.

8. Firebase. *Github.* [Online] https://github.com/firebase/firebase-functions.

9. Google. *Firebase.* [Online] https://firebase.google.com/docs.

10. Redux. *Wikipedia.* [Online] https://en.wikipedia.org/wiki/Redux_(JavaScript_library).

11. Material UI. [Online] https://material-ui.com/.

12. Design Patterns. *Source Making.* [Online] https://sourcemaking.com/design_patterns.

13. **Katie James Watkinson, Alexis Dent**. *BRIDES.* [Online] November 20, 2019. https://www.brides.com/story/5-steps-to-wedding-budget.

14. **Forrest, Kim. *WeddingWire.*** [Online] June 5, 2019. https://www.weddingwire.com/wedding-ideas/wedding-vendor-list.