

'Pleisure' Platform

Software Description

Pleisure is a web platform which connects parents with activities providers for their children. It aims to offer parents the capability of finding high quality activities for their kids via a simple and easy to use user interface.

At the same time, it provides a handy environment for corresponding services and businesses for advertising and reaching out to more potential customers.

Requirements Analysis

Functional Requirements

User Groups and corresponding operations

The users of the platform can belong to one of the following groups:

1. Guest users
2. Parents
3. Organizers
4. Administrators

Guest users

Permissions: Guest users have access to most of the content of the platform. They can search and view available events as well as apply filters, to limit the displayed results to their preferences.

Restrictions: Guest users do not have a profile, as a result they are not allowed to have a wallet or buy credits. Guest users cannot create or book events. They also do not have any administrator privileges.

Parents

Permissions: Users signed up as Parents can perform all the actions that a Guest user can. Additionally, they have access to a private profile page where they have their personal and contact information, photos, booking history and wallet with their credits (see [Monetization](#)). Parents can buy credits with any of the given payment methods, and they can use those credits to book events. In this personal page there is also an option to "add child", via which, a parent can add information about his children (such as date of birth and preferred activity), so that the events displayed in the search results will have pre-applied filters accordingly for quicker personalization.

Restrictions: Parents cannot create events and they do not have any administrator privileges.

Organizers

Permissions: Organizers are the activities/events providers. They can perform all the actions that Guest users

can. In addition, they have a public "profile page" in which they have the "add event" option, where they can create an event, by supplying the necessary information. These information include the exact time and date of the event, info about it being recurring or not, its pricing, address (geolocation), photos etc. Like parents they can use the "report" function to mark any inappropriate activity. Finally, they can also view monthly reports in their profile page, with all the completed events, total respective number of tickets sold and total income for each event in EUR.

Restrictions: Organizers do not use the credits system, so they do not have a wallet. As a result cannot book other events. They also do not have any administrator privileges.

Administrators

Permissions: Administrators are special users that can manage the registered users (Parents and Organizers). They can:

- Ban users, if they notice any violation of the Terms of Usage
- Restrict user functionality (such as event creation for an Organizer)
- Edit passwords

Restrictions: They do not have access to any private information of the users, such as passwords, billing information etc.

Authentication of all users (belonging to the groups Parents/Organizers/Administrators) is performed with the use of passwords that are set on each respective signup.

Entry Points

All the users visiting the web platform, can immediately utilise the functionality of the service by typing in the type of activity in the main textfield.

The nearby events with matching tags are then displayed on another page, to which the user is redirected after performing the query action.

Else, by clicking the "Events" tab on the navbar they are getting redirected directly to the event page, where they can browse events and apply filters.

Monetization

The platform uses an internal currency system.

More specifically, different types of users can interact financially with each other by using "credits". "Parents" can buy credits via PayPal, and use these tokens to pay the "Organizers" for the chosen activities.

The amount of credits is bound to the user's account, and no further kind of transactions are supported. Also, refunds and cancelling of transactions are not supported.

Moreover, the website's owner can profit by keeping a commission for every transaction fulfilled.

Booking events

Events can be booked by parents with the use of 'credits'. Upon booking the event, the ticket is presented and then emailed to the corresponding 'parent' user as a pdf file.

Event Search and Results Filtering

As mentioned before, 'Pleasure' offers a simple search engine on the website's main page. This aims to enable every user, to narrow the list of given events, so that it matches their needs and preferences. The user types in an activity and the corresponding events are shown immediately without having to browse through all website's events.

'Pleasure' provides further result filtering on the Events page. In order to use the filtering tool a location must be provided by the user. Given that, the user limits the event shown results by adjusting:

- Distance from the given location
- Maximum Price of the event
- Age Range of the participants
- Activity Category
- Date of the event

The events are displayed in two ways, either of which can be active at any time by selecting the corresponding tab. They can be displayed in the form of a scrollable list, or in a map, as marked points which can be examined and/or clicked upon for redirection to the event page for further details and booking.

Non-functional Requirements

Legal

Software license

The software is licensed under the MIT License:

Copyright (c) 2017 Pleasure

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Cookies Policy

Users can instruct their browsers to refuse all cookies or to indicate when a cookie is being sent. However, if they do not accept cookies, they will not be able to use some portions of the service.

Security

The software implements all the security standards to prevent SQL injection attacks, XSS, and user accounts leak from the database. Sessions are over TLS so all communication with the server is encrypted. The sensitive data of the users (such as information about the children of the "parent" users) are visible only to the user himself and the Administrator users.

Furthermore, since the application is going to handle a currency-like system as well as paid event booking, security coherency and fault tolerance are also big concerns.

Usability

The User Interface is the simplest possible, so that even inexperienced users can have full experience of the functionality of the service. It features easy navigation for all user groups.

Flexibility

The software can be deployed on any platform that can run .NET >= 4.6 or Mono >= 5.4 and a MySQL database.

It can easily be managed through an Administrator account with no need for technical skills or low-level access.

Deployment

The installation procedure (both building and deployment) will aim to be as automated as possible with the use of a script.

Responsiveness

The UI is responsive, meaning it offers optimal viewing experience across a wide range of devices (Desktop, Tablet, Mobile).

The forms and interactions with the page will utilize the asynchronous properties of web-development languages, so that time-consuming tasks can run in the background without making the page unresponsive.

Technical Specifications

Architecture and Design

The development stack has been arranged as follows:

- Database: **MySql**
 - The relational model, generally suits well any application that does not work with any Artificial Intelligence technologies, or any form of data that is otherwise sparsely defined and/or

distributed. Therefore, it became apparent that the application's back-end storage would be within the SQL family. Within that set of languages, MySQL was pretty straight-forward choice as it is free, flexible, easy to set up or use and widely supported in any programming language by numerous libraries and functions.

- Web-Server: **C#**
 - Although the choice here was mostly preferential, it is a strong one, as the .NET Framework is very versatile, robust and agile in development. This language is further supported by Microsoft's own Visual Studio, which makes any workflow effortlessly streamlined and optimized.
 - For any interaction between the web server and the application, the former will provide a JSON API for data retrieval, as well as any operation the user might intend to perform through the application, for example booking an event or changing their password.
- Front-End: **HTML, CSS, Javascript**
 - As this is a web application, the default W3C set of languages has been chosen to both design and render the application. Any alternatives would involve the use of specialized front-end rendering frameworks, which were deemed to be beyond the scope of the project. Describing the application natively in the web languages also ensures abstraction, making the front-end of the project not have any major dependency on any other parts of stack.

Interfacing

For static information, the server will use session variables to properly render an html page. For example, the following HTML snippet will adjust what is displayed in a particular part depending on whether the user is currently logged in or not.

```
@if{is_loggedin}  
    <h4>Hello user!</h4>  
@else{  
    <h4>Hello guest!</h4>  
@endif{}
```

And the following snippet will display a greeting to the user, as well as their avatar.

```
<span>Hello @{user.name}!</span>  

```

For data that is either not part of the page itself or that needs to be dynamically fetched and displayed, a private HTTP API will be queried by the front-end using AJAX. The server's API will respond to the following URLs:

Endpoint	Method	Description
/login	POST	Attempt to login a user
/register	POST	Attempt to register a user

/signinout	POST	Sign out the user and destroy his session
/api/events	GET	Get a list of all events matching certain criteria
/api/kids	GET	Get a list of the kids of the currently logged in user.
/api/email_available	GET	Check to see if an email is available for registration
/api/user_update	POST	Update a user's info.
/api/event_create	POST	Called by an organizer to create a new event.
/api/book_event	POST	Called by a parent to book an event for one of their kids.

Additional endpoints will most likely be added during the rest of the development for other minor functions.

Creating and booking events

The primary function of the application will be booking spots at events. As such, the system will be designed as follows:

Organizers will create **Events**, which will be defined by the following:

- Title
- Location
- Duration
- Minimum and maximum age
- Price
- Maximum number of attendees (*optional*)
- Accepted genders (*optional*)

An organizer will then have to **Schedule** their events through their private profile, by specifying:

- Date
- Time
- Recurrence (*optional*)

As such, an **Event** will remain in the database even after it has been concluded and can be scheduled again, or even multiple times.

An **Event** that has no upcoming scheduled date will not appear in search results.

Aa **Event** can be cancelled by the organizer. In such cases the parents will be notified via e-mail and the event's price in credits will be immediately refunded in full.

Parents will *have* to add their **Kids** to their private profile by defining:

- Name
- Age
- Gender

Therefore, a parent booking an event for one of their kids will add the kid to the list of attendees for that

Event. Obviously, in order for a kid to attend an event, it needs to comply with the age and gender requirements of the event.

The parent will be allowed to choose which scheduled date and time of an event they are booking for. In the case of recurring events, only the immediate occurrence can be booked.

Frameworks / Libraries

Back-End

- [Newtonsoft.Json](#)
 - The choice of this library has become pretty obvious when it comes to web applications that communicate over APIs. In our case, the communication between the web page and the web server is done via asynchronous HTTP requests and all the messages are in JSON format, thus making this library almost necessary.
- [HaathDB](#)
 - This library extends - and depends on - the official [MySQL.Data](#). Offering significant simplification and abstraction over basic database functions, through serialization, this library - in most cases - eliminates the need for most additional development when integrating C# with MySQL.
- [Watermark](#)
 - The other part of this project, a library that will be used to watermark images provided by event organizers. Its function will be to protect them from copyright theft, which can happen on a platform such as this one.
 - The releases of this project are hosted on a [private NuGet server](#).
- [Chance.NET](#)
 - This library will not see any use during the deployment of the application. It's a randomness generation which will be used during development, by generating large amounts of fake data for testing purposes.

Front-End

- [jQuery](#)
 - This leading library in DOM parsing and manipulation, will be utilized for just about every function of the website. Using its AJAX capabilities for asynchronous http requests, as well as its high-level methods for document event handling, each and every page will be designed to be fast and responsive.
- [Bootstrap 3](#)
 - Bootstrap will be used mostly for designing the responsive layout of the pages but also for providing certain functionalities, like pop-up windows.

Wireframes

The wireframes describing the user interface, for the most basic pages of the platform are shown below:

Home Page:

Pleasure

Events

Log In - Sign Up

Pleasure

Button

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Events Page:

Pleasure

Events

Log In - Sign Up

Filters

Filter

Nearby Events for your children!

ListView

MapView2

Panel

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Panel

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Panel

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Panel

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Event Page:

Pleasure
Events
Log In - Sign Up

Event Title

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Details

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Attend

Map

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Profile Page:

Pleasure
Events
Log In - Sign Up

Name Surname

Label
Label
Label

My Kids / My Events
+ Add Kid / Host Event

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Completed Events

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Label
Label

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

Admin Panel:

Pleasure
Events
Log In - Sign Up

Admin Panel

User Management Panel

id	active	name	email	role	edit
1	<input checked="" type="checkbox"/>	takis	takis@pleasure....	admin	<input type="button" value="edit"/>
2	<input type="checkbox"/>	makis	makis@pleisur...	organizer	<input type="button" value="edit"/>
3	<input checked="" type="checkbox"/>	sakis	sakis@pleasure...	parent	<input type="button" value="edit"/>

Lorem ipsum dolor sit amet,
 consectetur adipiscing elit.
 Nunc maximus, nulla ut
 commodo sagittis, sapien dui
 mattis dui, non pulvinar
 lorem felis nec erat.

Lorem ipsum dolor sit amet,
 consectetur adipiscing elit.
 Nunc maximus, nulla ut
 commodo sagittis, sapien dui
 mattis dui, non pulvinar
 lorem felis nec erat.

Lorem ipsum dolor sit amet,
 consectetur adipiscing elit.
 Nunc maximus, nulla ut
 commodo sagittis, sapien dui
 mattis dui, non pulvinar
 lorem felis nec erat.

For other functionalities (LogIn/SignUp, Add Event, Add Kid, Add Funds etc.) we use modals (popup dialog boxes), and not dedicated pages in order to avoid many redirections and keep the UI simple.

System Requirements

Environment

The web server can run in either one of the following environments with their respective dependencies

- Microsoft Windows
 - .NET Framework 4.6 or higher
- Any unix distribution
 - Mono Framework 5.4 or higher
- Any unix distribution
 - Docker

The web server can be built in either one of the following environments with their respective dependencies

- Microsoft Windows
 - NuGet
 - MSbuild
- Any unix distribution
 - NuGet
 - xBuild
- Any unix distribution
 - Docker

The application also requires a stable connection to a **MySQL server**.

Hardware requirements

The application itself has a small memory footprint and is linearly scalable. The application has to store the high-resolution images that the organizers will upload and thus has a significant requirement for storage.

To ensure better scalability, events that have concluded will be deleted from storage if they are not scheduled again after a certain amount of time.

Build & Deployment

The project's deployment will revolve around [Docker](#). A linux bash script will be used to automate the process, which will do the following:

1. Pull the source code from the repository.
2. Build the project using a Makefile which in summary does the following
 1. Add <https://nuget.gmantaos.com/api/v2/> to a temporary NuGet source, since some of the dependencies are hosted there.
 2. Use `nuget restore`, which will download all the dependencies of the solution.
 3. Remove the above temporary source.
 4. Use `xbuild`, which is a clone of the official `MSBuild`, to build all projects in the solution in the correct order using the *Release* configuration.
3. Build the docker image from the `Dockerfile` which in summary does the following
 1. Use an image that has `mono-complete` installed.
 2. Copies the build folder `Pleasure/bin/Release/` to `/opt/Pleasure` inside the container
 3. Runs the web server as the image's entrypoint using the command `mono /opt/Pleasure/Pleasure.exe`

Additionally, any options that need to be passed to the application, such as the credentials for the MySQL server for example, will be passed when running the container in the form of environment variables, like the following example

```
docker run \  
  -e MYSQL_HOST=172.17.0.1 \  
  -e MYSQL_USER=progtech \  
  ...
```

Monitoring

As the application will run in a Docker container, there are plenty of ways to monitor its status and performance

The following command will follow the stdout of the web server running inside the container

```
docker logs -f pleasure
```

The following command will provide diagnostic information on the running container

```
docker stats pleasure
```

CONTAINER	CPU %	MEM USAGE / LIMIT	MEM %
NET I/O	BLOCK I/O	PIDS	
pleisure	0.00%	95.36MiB / 923.4MiB	10.33%
10MB / 11.5MB	58.5MB / 1.62MB	0	

Notes on security

- Storage of the passwords in the database is done after hashing them with 'salt' to prevent any leak (in case of database attack)
- The server can run over TLS and a certificate can be provided.