**Chapter 1: Introduction**

The Company “E3T” was founded a couple of years ago by three friends. After slight disagreements, the group was disbanded and E3T is currently run by CEO Mr. Gerjan van Oenen. He hopes to be the bridge between the citizens and talents all over the Netherlands. The company currently has 15 workers responsible for jobs such as marketing, sales, finance, logistics, etc.

All the work was done and recorded onto sheets of paper. This method of documentation required labor as it was harder to document. This hard labor led them to collaborate with the company of PECZARB, with the newly combined workforce, E3T is leading to take a big leap in the internet world where all the events and organizations will be recorded and tracked online. Upon completion of the project, E3T will achieve its goal of having their work done both faster and automatically.

With the newly designed project, they plan to finally connect their clients to the talents. With the introduction of this new information system, most of the work will be handled by the server which also displays all the events. Now, rather than filling out paper forms, the workers will be able to work more efficiently by documenting all the events online.

What we call a functional design can simply be described as a tool to simplify the design of the project. It described the hardware and software used. It also serves the purpose of separating each component, so they don’t intervene with one another and work in harmony.

**Chapter 2: Description of the new system**

The Current project “WTS” developed by PECZARB has basic working principles. In simple terms, it can be described as a social media platform for new talents to be discovered by people. It works like most of the social media platforms currently available for the public or a platform for people to follow the events organized by E3T.

As for the citizen’s side:

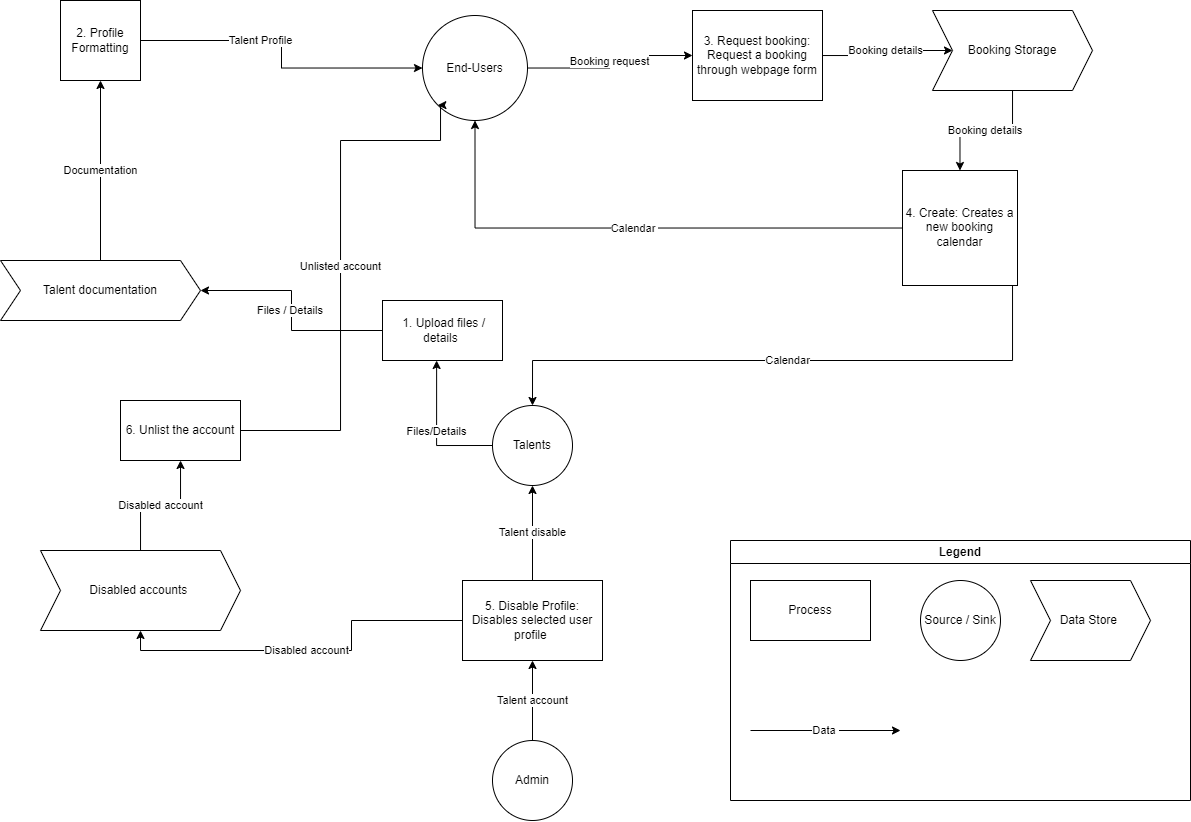
Users will be able to see all the different events scheduled. If they wish to, they can sort individual talents via the search bar, for example, if they type “clown”, they will be able to access all the clowns available for a show. Afterwards, they may choose an appropriate day for the event of their choice.

As for the talent’s side:

All talents will be provided with an account, and they will be verified by E3T. After verification, all artists will have their page where they can upload pictures and documents. They can display information about themselves or when they are free for an event. They will be able to display their working hours, which days they are booked and what activities they can perform. They will also have the option to show themselves as “available” or “unavailable” depending on their timetable.

The E3T side:

Some workers of the company will be given administrative roles, where they can block accounts that break TOS. Admins will be able to access every member’s pages. They will also have the administrative ability to disactivate people’s accounts depending on their actions; they can do this by marking the accounts as “inactive”.



**Chapter 3: Data model**

|  |  |  |
| --- | --- | --- |
| **Name** | **User** | |
| Description | Can either be an admin or a talent. This will be used for the login page | |
| Number of records | From 1 to 17 million | |
| **Attributes costumers** | **Description** | **Type** |
| idUser | Unique code to identify the user | Int, Auto Increment, Primary Key |
| dtName | Name of the user | Varchar 50 |
| dtLastName | Last Name of the user | Varchar 50 |
| dtNumber | Their telephone number | Varchar 20 |
| dtEmail | That person’s email address | Varchar 250 |
| dtPassword | That person’s password hashed | Varchar 300 |
| dtIsAdmin | 1 or 0 to find out if that person is an Admin or a Talent | Boolean |
| dtRating | Rating of the talent | Float |
| dtImg | They can have a picture of themselves, so that the clients know who they are working with | Varchar 50 (can be empty) |
| fiSpecialty | Linking what the talent’s specialty is | Int |

|  |  |  |
| --- | --- | --- |
| **Name** | **tblBooking** | |
| Description | This will display which Talent is booked when and the email of the client that booked the talent | |
| Number of records | From 1 to 17 million | |
| **Attributes costumers** | **Description** | **Type** |
| idBooking | Unique code to identify the booking | Int, Auto Increment, Primary Key |
| dtDate | Date of the booking | DATETIME |
| dtTypeOfBooking | Here the client can specify what the event is going to be (a party, a wedding, etc.) | Varchar 50 |
| dtEmailClient | The email of the client who booked the talent | Varchar 250 |
| fiUser | Id of the Talent, so they are linked to the booking | Integer, Foreign Key |
| fiSpeciality | What booking it’s for | Integer, Foreign Key |

|  |  |  |
| --- | --- | --- |
| **Name** | **tblUnavaible** | |
| Description | Here is where the Talent will say if they are available or not as they can be on vacation, dealing with personal issues, etc. | |
| Number of records | From 1 to 35 million | |
| **Attributes costumers** | **Description** | **Type** |
| dtDateStart | Start of the Talent’s unavailability | DATETIME, PRIMARY KEY |
| dtEndDate | End of the Talent’s unavailability | DATETIME, PRIMARY KEY |
| fiUser | Id for the Talent that is not available | DATETIME, PRIMARY KEY |

|  |  |  |
| --- | --- | --- |
| **Name** | **tblSpecialty** | |
| Description | Name of the talent specialty | |
| Number of records | From 1 to 50 | |
| **Attributes costumers** | **Description** | **Type** |
| idSpecialty | Id of the specialty | Primary key auto inc |
| dtDescription | What is the specialty is it singing, dj etc | Varchar (30) |

**Chapter 4: Desired Output**

Table 1: Output product for talents

|  |  |
| --- | --- |
| Code of the product | **Req-24.1** |
| Name of the output | Personal page for talents (public or private) |
| User | Admin  Talent  General User |
| Aim | To allow talents to create a personal page with information pictures and videos and allow them to decide what is visible by the general users. |
| Frequency | Per page request |
| Sorting | Uploaded files are sorted by time and date |
| Selection | Talent Name  Contact info (Email, phone number, address)  Unique ID |
| Data to be printed | Talent’s profile picture  Talent’s name  Talent’s contact information  About me section  Files uploaded by a talent  Captions for uploaded files |

Sketch 1: Personal page for talents as an information sketch

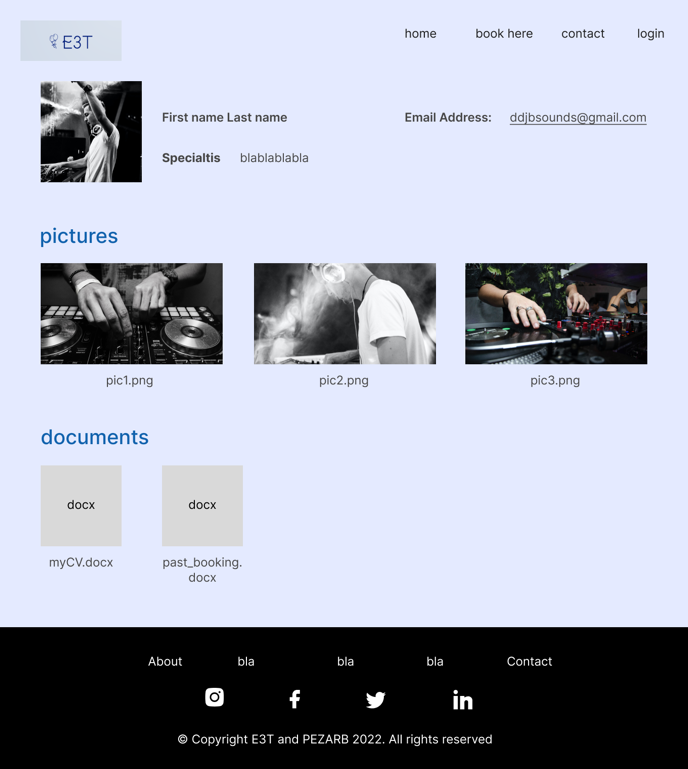


Table 2: Calendar Output product

|  |  |
| --- | --- |
| Code of the product | **Req-24.2** |
| Name of the output | Calendar (booking options) |
| User | Talents  General User |
| Aim | To show general users when a specific talent is available and when they are not. Also, to allow talents to mark when they are available or unavailable. |
| Frequency | Per page request |
| Sorting | Date |
| Selection | Date |
| Data to be printed | Year  Month  Name of days (e.g., Monday Tuesday...)  Days numbered  Coloring of dates based on meaning |

Sketch 2: Calendar information sketch

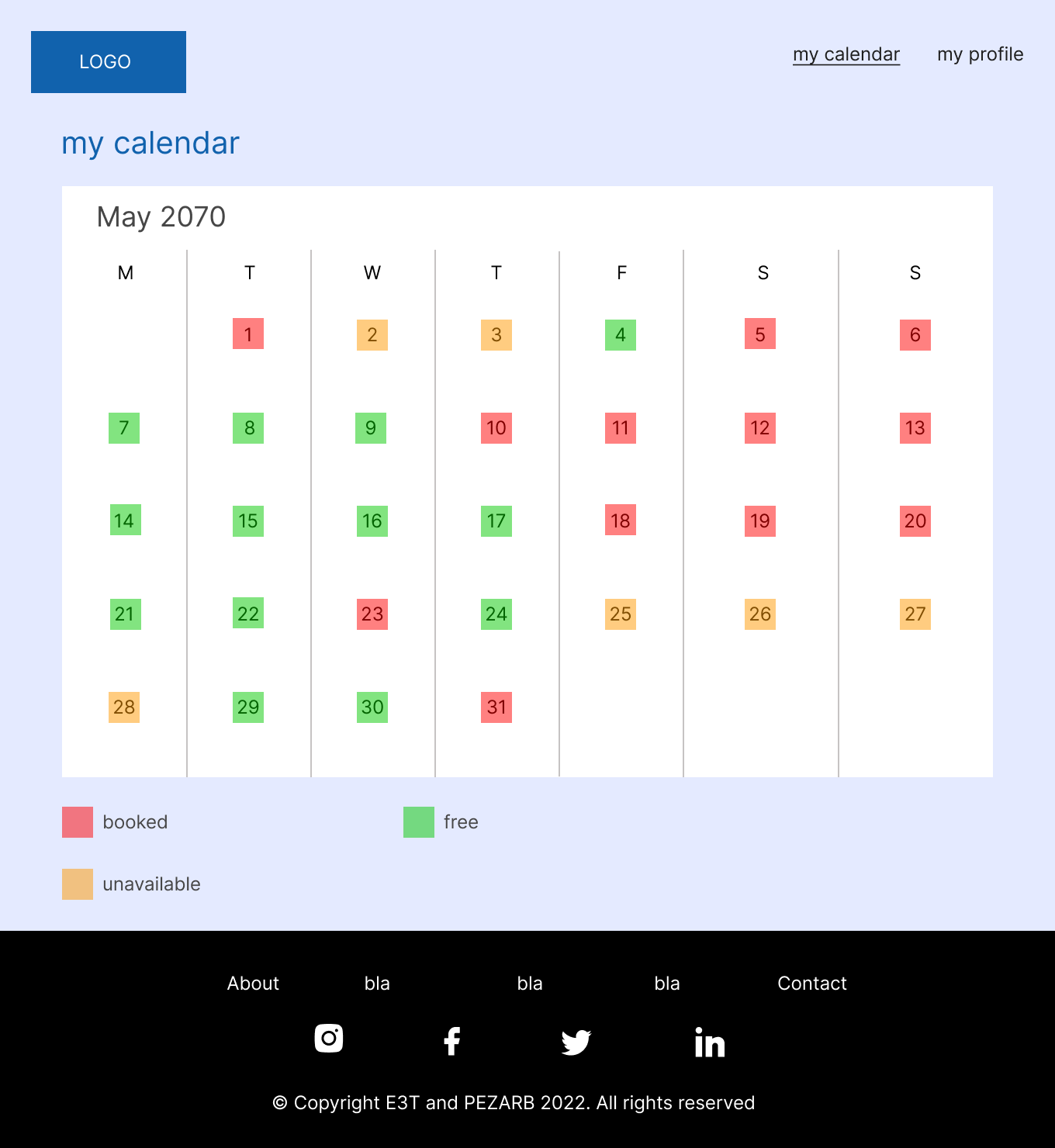


Table 3: Talent search and list function as product

|  |  |
| --- | --- |
| Code of the product | **Req-24.3** |
| Name of the output | List of talents |
| User | General user |
| Aim | To provide a full list of (available and unavailable) talents and to allow the search between those talents based on tags, name, or price. |
| Frequency | Per page request |
| Sorting | Availability (available now -> available later)  Name  Talent type (e.g. Dancer, Guitar, Piano, Singer)  Tag |
| Selection | Available now  Available for the next week  Available for the next month |
| Data to be printed | Talent’s profile picture  Talent’s name  Talent’s contact information  Talent Type  Tag(s) |

Sketch 3: Talent search and select functionality as an information sketch

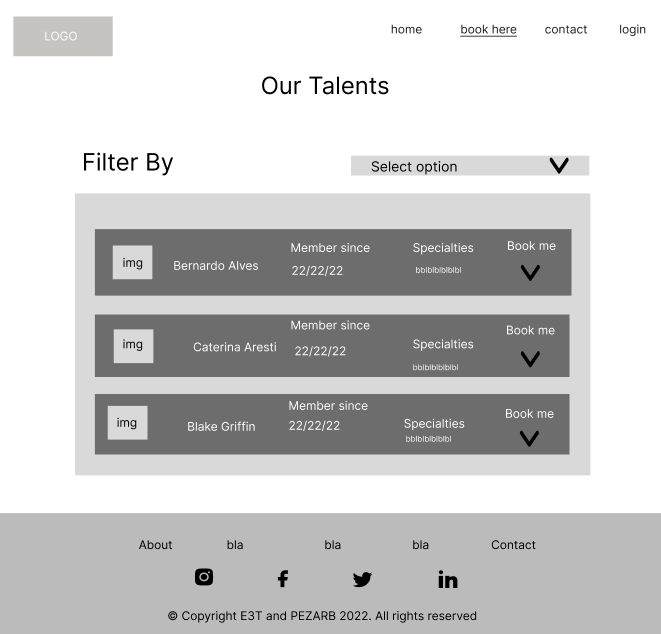
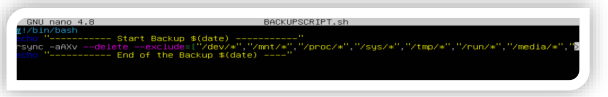


Table 4: Back up for front end and back end as product

|  |  |
| --- | --- |
| Code of the product | **Req-28** |
| Name of the output | Back up for system and front end |
| User | Admin |
| Aim | To have an additional layer of security in a form of storing (actual up to this point; previously saved) data on a different machine, so that in a case of unexpected data loss, it would be possible to restore database(s) and / or any other file(s) to stable version. |
| Frequency | Daily |
| Sorting | Date |
| Selection | Date |
| Data to be output | Database(s) and / or various important files |

Sketch 4: Backup for front-end and back-end as a functionality



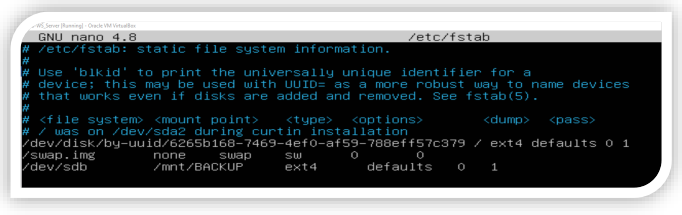


Table 5: Specified admin functionality as end product

|  |  |
| --- | --- |
| Code of the product | **Req-10.1** |
| Name of the output | Specified Admin Functionality |
| User | Internal stakeholders |
| Aim | To allow the internal stakeholders to govern their own community by giving them access to specified functionalities in the website. This could include adding talents, deleting talents, activating or deactivating them as well. |
| Frequency | Per page request |
| Sorting | User ID  Email Password |
| Selection | User ID |
| Data to be printed | Activate  Deactivate  Delete profile  Add profile |

Sketch 5: Admin specific functionality as an information sketch

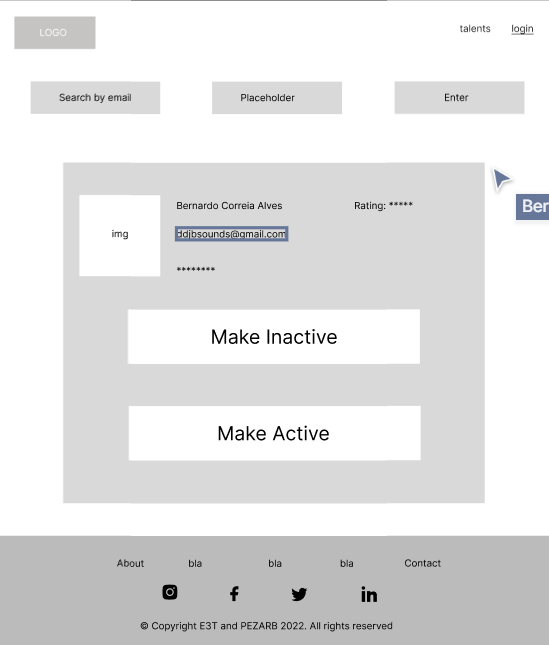


Table 6: Specified Talent Functionality as end product

|  |  |
| --- | --- |
| Code of the product | **Req-10.2** |
| Name of the output | Specified Talent Functionality |
| User | External stakeholders (Talents) |
| Aim | To allow the Talents to access their own account and further specified functionalities based on their credentials (email, passwords) |
| Frequency | Per page request |
| Sorting | User ID  Email  Password |
| Selection | User ID |
| Data to be printed | Personal page  Available for booking  Busy (booking function) |

Sketch 6: Talent specific functionality as an information sketch

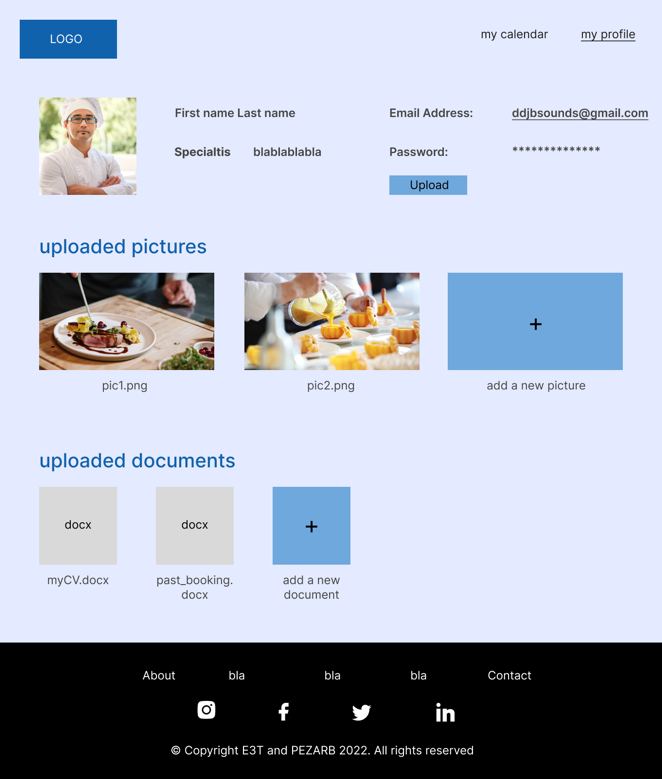
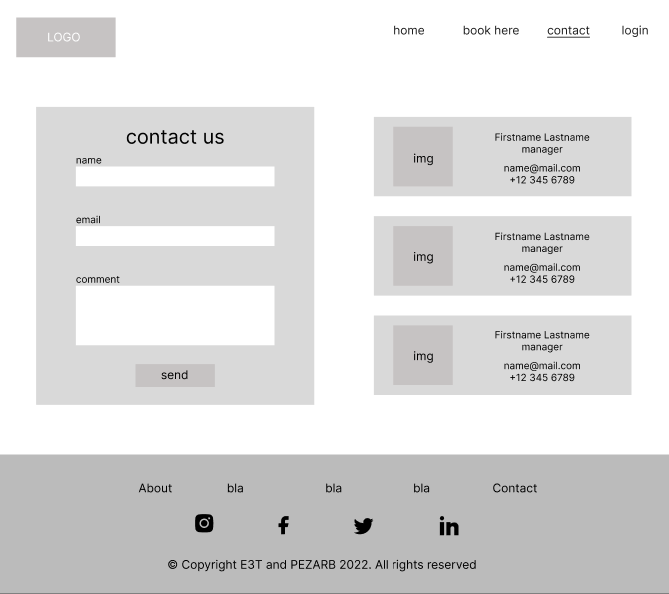


Table 7: Specified user functionality as end product

|  |  |
| --- | --- |
| Code of the product | **Req-10.3** |
| Name of the output | Specified User Functionality |
| User | General user |
| Aim | To allow the users to browse the website, the talents and their availability with the option for booking them. This needs to be available without the need for an account or credentials. |
| Frequency | Per page request |
| Sorting | Email |
| Selection | Email |
| Data to be printed | Sorting of talents  Booking talents |

Sketch 7: General user functionality as an information sketch



**Chapter 5: Required Input**

Table 8:

|  |  |
| --- | --- |
| **Code input task** | **Input task IT-012** |
| Name | Entry of bookings |
| Authorization | Customers and E3T |
| Objective | Registering bookings |
| Description | For a customer to make a booking they need to provide an email address and select the date for the event.  Each booking, the system automatically registers the data into the database.   * The customer inputs his email * The customer sets the date * The system checks if the talent is booked * If the talent is not booked, the booking will then be confirmed . |
| Frequency | Every booking |
| Files | Booking   date and time                      Add record  Customer email                Add record  Talent calendar                 Record is read |
| Screens used | Main screen  Talent booking screen |

Table 9:

|  |  |
| --- | --- |
| **Code input task** | **Input task IT-013** |
| Name | Talent |
| Authorization | Admin |
| Objective | Register Talents |
| Description | The admins need to be able to register talents manually into their database in order to allow them access to specified functionalities. They will be able to access their account after using the Login functionality with their email and a specified password. The following information needs to be input by the admin to register a talent:   * Full Name * Email * Phone number * Specific talent |
| Frequency | Every Talent application |
| Files | User ID ---- add record  User email ---- add record  User password ---- add record  Phone Number ---- add record  Speciality ---- add record |
| Screens used | Log In page  Personal talent page |

Table 10:

|  |  |
| --- | --- |
| **Code input task** | **Input task IT-014** |
| Name | Loging in |
| Authorization | E3T and talents |
| Objective | Accessing the data linked to the account |
| Description | Admin log in:  After admins are provided with credentials by the PECZARB team they can log in the “log in” page.   * PECZARB provides the credentials * The admin logs in their account * The admin now has access to their account profile and all the functionalities of the admin role   Talent log in:  After one of the administrators creates the account for the talent, the talent can log in at the “log in” page with the credentials provided to access their account.   * The administrator provides the credentials * The talent logs in their account * The talent now has access to their account profile |
| Frequency | Every time the talent accesses the system |
| Files | Talent account                    Record is read  Admin account                 Record is read |
| Screens used | Main screen  Log In screen |

Table 11:

|  |  |
| --- | --- |
| **Code input task** | **Input task IT-015** |
| Name | Admins |
| Authorization | Admins |
| Objective | Adding further admins to the server |
| Description | Admins should have the right to make other users admins or to add entirely new users as admins. This is so that if they need help running and organizing the website they can simply and effectively add new admins who have the same rights as them. Admins will need the following information in order to create a new admin:   * User ID * Email * Password * Admin value (boolean) if it is true than they are an admin, if not then talent |
| Frequency | Every Admin addition |
| Files | User ID ---- add record  User email ---- add record  User password ---- add record  Admin or not ---- add record |
| Screens used | Login page  Admin page |

Table 12:

|  |  |
| --- | --- |
| **Code input task** | **Input task IT-016** |
| Name | Rating |
| Authorization | General users |
| Objective | To allow general users to rate the talent |
| Description | The general user should be able to rate a talent which then should be displayed on the talents personal page. This should be done by averaging all of the inputs from the users, and then displaying that average. |
| Frequency | Every rating |
| Files | Talent ID ---- add record  Rating 1, 2, 3, 4, 5 … ---- add record  Average rating ---- processed from added record |
| Screens used | Talent page |

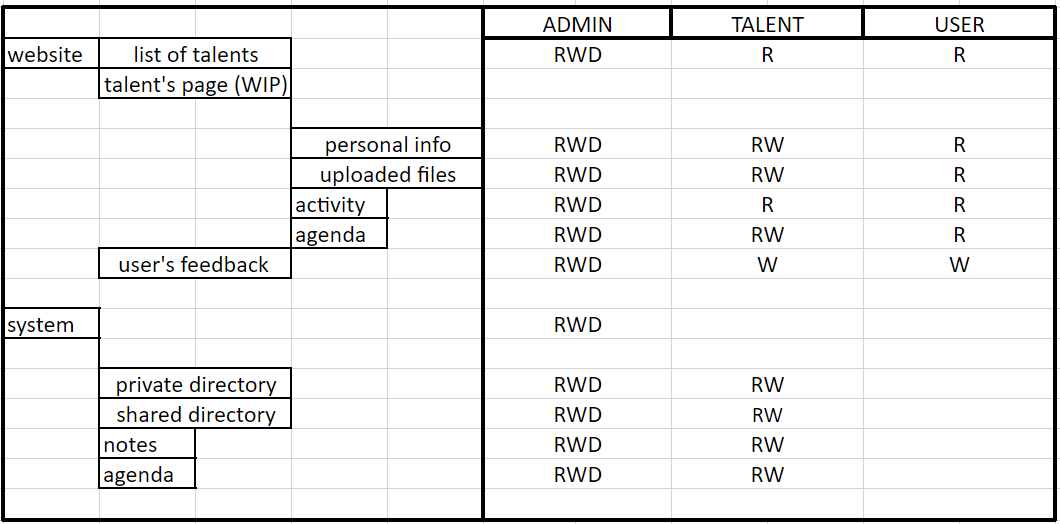
Table 13:

|  |  |
| --- | --- |
| **Code input task** | **Input task IT-017** |
| Name | Uploading pictures and data |
| Authorization | E3T and talents |
| Objective | Uploading pictures and data to the account profile |
| Description | After the talent logs into their account, they can upload data on their profile. |
| Frequency | Every time the talent uploads data |
| Files | Talent account                    Record is read  File                 Add record |
| Screens used | Main screen  Log In screen  Profile account screen |

Table 14:

|  |  |
| --- | --- |
| **Code input task** | **Input task IT-018** |
| Name | Selecting specialty |
| Authorization | E3T and talents |
| Objective | Providing the client with more information about the talent |
| Description | After the talent logs into their account, they can select a speciality to display on their profile. |
| Frequency | Every time the talent selects a specialty |
| Files | Talent account                    Record is read  Specialty               Add record |
| Screens used | Main screen  Log In screen  Profile account screen |

**Chapter 6: Menu Structure and Authorization**



**Chapter 7: Organizational consequences**

**How will the system developed be tested?**

The system will undergo performance testing, recovery testing, security testing, alpha testing, and beta testing.

Performance testing: Internal testing will be done on the system's runtime performance. We'll evaluate the system's scalability, responsiveness, and reliability.

Recovery testing: We'll intentionally cause the system to crash in order to check the viability of the data backups and the automaticity of the system's recovery.

Security testing will be done to identify any weaknesses in the security measures put in place and to make sure the system is protected against erroneous intrusions and injections.

Alpha-testing: While the system is being developed, internal tests will be run on it to look for any errors or bugs and make sure the functionalities implemented function as intended. In this phase, unit tests and integration tests are written and used. The developed system will undergo beta testing after numerous alpha tests.

For the acceptance testing, the system will be deployed in a test environment, it will be made available to users, Feedback will be collected from users and any errors/bugs will be documented. Changes will be made to the system and the system will be delivered to production.

**How will the acceptance by the users of the new system be arranged?**

After development, A date will be selected for acceptance testing, the system will be made available in a test environment to end-users to see if our system fulfils all user requirements. This exercise will also help to detect any errors/bugs in the system.

**How will the conversion be performed; therefore, how will the change be made from the old situation without the information system to be developed– to the new situation? How will the old data be entered into the new system?**

Data was previously stored on paper, the new system's database will be created with the structure of how data was previously stored on paper in mind; after the database is created, data will have to be manually entered into the database.

**Which conversion problems can be expected?**

We expect problems such as:

* Double data entry
* Data entered stored with a wrong data type
* Corrupted/missing papers
* Missing primary & foreign keys

**Which training courses are required for the end users?**

The training courses for end users include:

* How to use the system
* How to use the web infrastructure
* How to add/remove talents from system (Admin)
* How to use the active directory

**Chapter 8: Technical Consequences**

**Are extra workplaces required, and if so, with which technical equipment?**

Yes, we will use 3 servers.

**Which other special** **technical equipment is required? For example, bar code**

**How many servers are used?**

There will be 3 servers for the Database, Webapp, active directory respectively.

**The system components are built on which OS?**

The webapp is built on Windows OS, the database and directory are built on Ubuntu.

**Are printers required?**

No, printers are not required, but they can be added to the system.

**Which data communication facilities are required?**

Data communication facilities required include;

* Routers
* Switches
* Ethernet cables
* Servers
* Computers

**Which internet facilities are required?**

Internet facilities required include;

* Email
* File Transfer Protocol
* World wide web (web app)
* DHCP
* Firewalls (IPtable)
* Active directory server

**Which security measures are used to protect data?**

Firewall (System), Password (Users), Backups (database& directory).

**Are extra security measures required to protect data?**

Password hash, Input validation, SQL injection protection.

**Are separate backup facilities required or can the backup be included with the other data from the network?**

Separate backup facilities are not required, the backups are on the same servers as the originals.

**Create a Configuration Plan**

Our project plan is our configuration plan.