families share

D2.1 Services design and deployment environment

WP2 – Platform Development

31.10.2018

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Disclaimer

Copyright message

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Document Information

Grant Agreement Number	780783	Acronym	FAMILIES_SHARE
Full Title	Families_Share mobile-based services definition		
Topic	H2020 CAPS Topic: ICT---11---2017		
Funding scheme	IA Innovation Action		
Start Date	1/1/2018	Duration	34 months
Project URL	www.families-share.eu		
EU Project Officer	Loretta Anania		
Project Coordinator	Agostino Cortesi, UNIVE		
Deliverable	D1.2: Families_Share mobile-based services definition		
Work Package	WP1 Conceptual framework for users' engagement, needs' analysis, behavior analysis strategy and co-design		
Date of Delivery	Contractual M10	Actual	29/10/2018
Nature	R – Report	**Dissemination Level**	P - Public
Lead Beneficiary	VILABS		
Responsible Author	Manolis Falelakis	Email	manf@vilabs.eu
	VILABS	Phone	+302310365185
Contributor(s):	All partners		
Keywords	Platform, architecture, deployment, development, specifications		

Document History

Version	Issue Date	Stage	Changes	Contributor
1.0	26 September 2018	Draft	Release of the first draft. Feedback on ToC.	VILABS
1.1	22 October 2018	Draft	Release of the second draft for feedback	VILABS
1.2	23 October 2018	Draft	Various improvements, mainly in the NFR descriptions, and compliance with ethics requirements	All partners
1.3	29 October 2018	Draft	Minor changes in ethics requirements and introduction	VILABS, FBK
1.4	29 October 2018	Final	Ready for submission	VILABS

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Executive Summary

This document reports the results of the activities conducted in work package 2 as part of Task 2.1 ("Setting the deployment environment") that aimed in specifying the functionalities and the software components that will be employed in the Families_Share platform, for the delivery of the selected services.

Taking into account the input from Task 1.3 and carefully considering the refined and prioritized set of user requirements and the related visual wireframes and flows specified in Deliverable D1.2 - Families_Share mobile-based services definition, this document provides:

- a list of the identified functional and non-functional software requirements for the first version of the platform.
- a description of the technological components that are going to be employed in the platform development and deployment.
- the design of the software architecture of Families_Share, both in terms of the software and its deployment in the CityLabs.
- the specification of the back-end Application Programming Interface of the platform, in the form of RESTful endpoints.

Moreover, the set of software requirements is revisited, specifying for each one of them how it is satisfied by the selected design.

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Table of Contents

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

1 Introduction

Families_Share WP2 aims to deploy the open source Families_Share platform, with the appropriate services, co-created in WP1, ready to be used by the pilots and evaluated with real users in WP3. In particular, the goals of WP2 are to:

•Leverage existing open source tools and an already existing open source platform (Cokido) and deploy a cloud based-software platform that support the sharing of caring and learning services among families.

• Provide localization of the service in the pilot countries languages.

• Test, validate and adapt the platform and the corresponding services.

• Deploy a final version of the platform that can easily be extended with additional services and replicated or scale up in other countries.

This document concludes and reports on *Task 2.1 – Setting the deployment environment*, the aim of which was to provide a final decision on the functionalities and the software components that will be used for developing the Families_Share platform. Work in Task 2.1 was carried out in close collaboration with WP1 and more specifically, Task 3.1, in order to ensure that the results of the first co-design activities carried out in the CityLabs are taken into account. The outcomes of Task 2.1 heavily affect the development activities of both the front-end and the back-end of the platform, which are being carried out in Tasks 2.2 and 2.3 respectively, while also providing guidelines for platform deployment, undertaken by Task 2.4.

Figure 1 summarizes the structure of WP2 Tasks and their relationship with Tasks from WP1 and WP3.

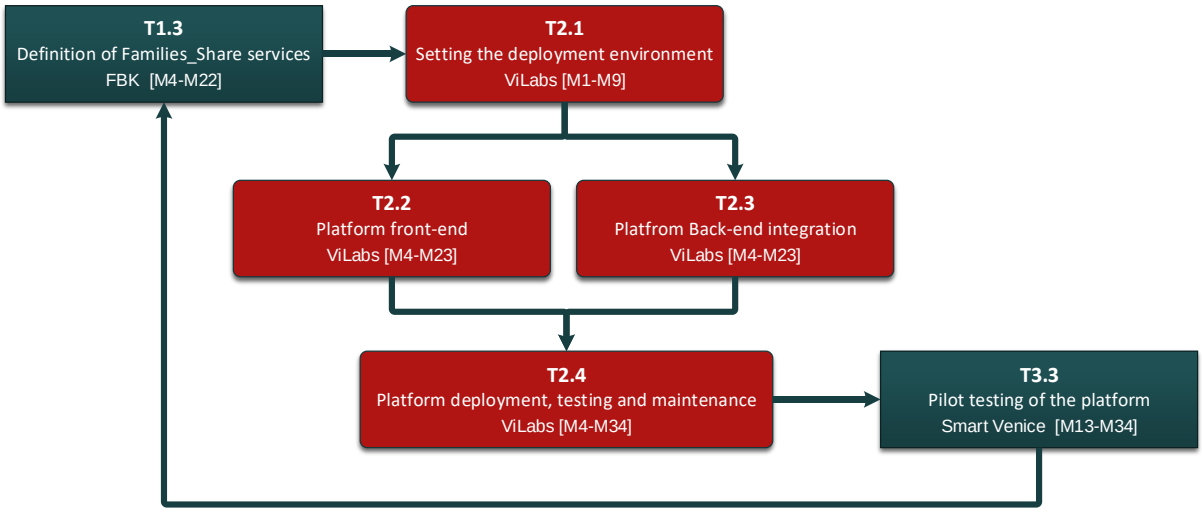

Figure 1: WP2 structure and relationships with T1.3 and T3.3

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

The platform deployment will occur in two phases that will be iterated two times, following the Agile development process. As shown in Figure 2, the first phase takes into account the co-design activities of WP1 and will be concluded in Month 14 with the delivery of the first version of the platform. This is to be evaluated in the first round of pilot activities carried out in the seven CityLabs. Taking place in a six-month period, from Month 15 to Month 20, this round of evaluation aims to involve a minimum of 100 users, i.e. parents of children aged 0-12 years. Results gathered in the first pilot testing will drive the next development phase of the platform, which will lead to the delivery of the final version in Month 23. The second version will be further evaluated during the second round of pilots, which will run from Month 24 to Month 32 and is planned to involve a total of 3500 real end-users within the seven CityLabs. Qualitative and quantitative methodologies (defined in WP3 and WP4) will be used to explore with parents and other stakeholders involved the strengths and weakness of Families_ Share according to the initial goals set by the project in WP1 (T1.2 and T1.3). Starting from the goals of the platform - raising awareness on social innovation and supporting networks of parents and citizens in organizing collaborative childcare - emphasis will be put on exploring both social and digital innovation components (e.g. usefulness, usability, efficiency, social impact).

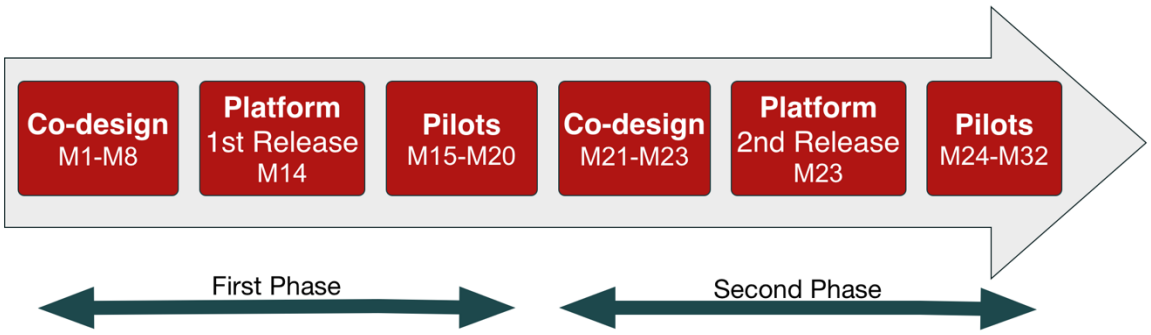

Figure 2: Iterative platform design process

1.1 Intended audience

This document targets Families_Share consortium members in order to reflect the progress of the implementation of the first release of the Families_Share platform. It also contributes to the overall dissemination of the activities and the results of the project.

1.2 Document structure

The document is organized as follows.

Section2 presents a listing of the identified functional and non-functional software requirements for the first version of the platform. Moreover, the technical aspects of the Cokido platform are described, along with the reasoning for the decision to not reuse its code.

Section 3 presents the technological components that will be used in the platform development and deployment.

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Section 4 specifies the design of the software architecture of Families_Share, both in terms of the software and its deployment together with the related back-end API endpoints. Moreover, the set of software requirements is revisited, specifying for each one of them how it is satisfied by the selected design.

Section 5 is devoted to technical compliance of the ethical requirements, as these were identified in WP7.

Finally, **Section 6** concludes the report.

2 Background

In the following sections we summarize the functional and non-functional requirements that will drive the platform development.

2.1 Functional requirements

The functional requirements of the platform have been extracted from the workshops in the participating CityLabs, captured in a structured form during the Venice co-design sprint in July 2018 and reported in Deliverable D1.2. Table 1 summarizing that list, is included here for completeness. The reader can refer to section 4.4 for a more formalized list of functional requirements, together with the way that each of them is foreseen to be satisfied by the platform.

Table 1: List of functional requirements.

ID	Requirement
FR-1	Website for each community
FR-2	Website shows the list of groups for that community (as in Cokido)
FR-3	The app has different instances
FR-4	App design is mobile first
FR-5	Search group
FR-6	Search users
FR-7	User name visibility
FR-8	User search visibility
FR-9	Cancel profile
FR-10	Suspend profile
FR-11	Export all information
FR-12	Integrate children information within the platform
FR-13	Showing children basic information + special needs
FR-14	Managing who can view your full children information

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

FR-15	Export children information
FR-16	Privacy consent at the registration
FR-17	Parents can add other users as parents
FR-18	User hierarchy
FR-19	New member notification
FR-20	New member acceptance rule
FR-21	Single sign on
FR-22	Community default options
FR-23	Community access option
FR-24	Group settings when creating a group
FR-25	Join a group with an invite
FR-26	Type of notifications
FR-27	Phone notifications
FR-28	Professional users
FR-29	Assign group roles
FR-30	Types of activities
FR-31	Promote groups and activity
FR-32	Evaluating an activity
FR-33	Sharing knowledge
FR-34	Starting page / Homepage for new users
FR-35	Starting page / Homepage for registered users
FR-36	Provide checklists
FR-37	Log-in with email
FR-38	Log-in with SMS
FR-39	Support communication within and outside the app
FR-40	Announcements (a.k.a. Messaging) feature
FR-41	Edit announcements / replies
FR-42	Deleting announcements / replies
FR-43	“Twitter” style notification

This project has received funding from the European Union’s Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

FR-44	Images in the announcements
FR-45	Activities are mapped in a collection of timeslots
FR-46	Timeslots have counters
FR-47	Propose different activities
FR-48	Problematic timeslots
FR-49	Export agenda as XLS
FR-50	Calendar view
FR-51	Toggle Calendar / List view
FR-52	Optimal solution
FR-53	Set constraints/ variables for slot confirmation
FR-54	Who can propose an activity
FR-55	Who can modify / confirm and close an activity
FR-56	Activities detail
FR-57	Timeslot information
FR-58	Professionals and timeslots
FR-59	Information on a timeslot
FR-60	Apply changes to activity time for all the time slots
FR-61	You can duplicate a time slot (for recurrent events)
FR-62	Export in your calendar
FR-63	Calendar on the homepage with commitment and fixed slots
FR-64	Calendar with colors

2.2 Non-functional requirements

This section summarizes at a high level the non-functional requirements of the Families_Share platform. Non-functional requirements define the quality attributes of a system and specify the criteria that judge its operation.

In particular, the emphasis is given on issues related to data privacy and security in order to ensure that all sensitive data (both personal and commercial) are protected. For this purpose, the industry standard web security practices and guidelines (e.g. data encryption) are taken into account for the design of Families_Share security components, with special focus on data security, availability, confidentiality and integrity of the applications and services.

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

The non-functional requirements are summarized in Table 2. The reader can refer to section 4.4 for a more formalized list of functional requirements, together with the way that each of them is foreseen to be satisfied by the platform.

Table 2: List of non-functional requirements.

ID	Requirement	Description	Category	Priority	Prototype version
NFR-1	Data encryption	The SYSTEM shall support encryption of sensitive data at rest and in motion. Sensitive data may refer to information about personal data and/or data generated by end-user behaviour. Managing and securing encryption keys is also important. Encryption keys should be stored separately from the data.	Security	High	v1
NFR-2	Backup	The SYSTEM shall be able to ensure the data integrity with frequent backups. All data will be replicated in remote systems in Athens and Thessaloniki in Greece. Data are backed up daily and a backup copy is stored offside but again in data centers in Athens and Thessaloniki in Greece.	Data Integrity	High	v1
NFR-3	Access management – User Authentication	The SYSTEM shall support authentication measures. Only authenticated platform users shall be able to login to the platform and use its resources except those specifically intended to be public. Authentication will be compliant with the European Parliament Directive 2002/58/EC.	Security	High	v1
NFR-4	Access management – Access Control	The SYSTEM shall provide an access control mechanism (e.g. role-based access control - RBAC) for restricting system access to authorized users (authorization). That is in order to provide the ability to differentiate users according to their roles in communities, groups and activities, as specified in the related functional requirements. In relation to API management, handling of exposed APIs including control of which interfaces are offered to which users is also foreseen.	Security	High	v1
NFR-5	Performance - Auditing	The SYSTEM shall support the capability of collecting logs for monitoring and auditing purposes.	Auditability	High	v1
NFR-6	Privacy – Anonymous Reporting	The SYSTEM must ensure that it is possible to provide anonymous reporting of analytics data such that the privacy of individual end-users is not compromised.	Auditability Privacy	High	v1
NFR-7	NFR-07: Personal data removal	The SYSTEM shall provide a data structure and a corresponding mechanism to allow users to completely remove their data, while ensuring the	Privacy Data Integrity	High	v1

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

		consistency of the remaining data.			
NFR-8	Sandboxed deployment	The SYSTEM will ensure that application instances, each corresponding to a separate community, will be deployed in a sandboxed manner, ensuring isolation from one another.	Deployment	High	v1
NFR-9	Multilingual support	The SYSTEM will ensure inherent support of multiple languages, with proper categorization and organization of the content, keeping language-specific resources separated from the rest of the application.	Localization	High	v1
NFR-10	Scalable deployment architecture	The SYSTEM will ensure that new application instances can be deployed easily, with no implication for the ones already running.	Deployment Scalability	High	v1
NFR-11	Data storage	Data will be stored at the central protected storage facility of a cloud service provider in Greece, administered by a responsible person from VILABS and certified data protection officer from the Cloud service provider and following the best practices and standards available. The Cloud service provider protected storage facility will be based on redundant systems and located in Athens and Thessaloniki in Greece. Data storage and protection procedures are compliant with 2016/679 (General Data Protection Regulation, GDPR) and European Parliament Directive 2002/58/EC.	Security	High	v1
NFR-12	Privacy Consent	The SYSTEM should support a privacy consent mechanism, denying access to users that do not provide it. In particular, children information are provided by parents and data are collected according to the GDPR (article 8). This is related to FR-12, FR-14, FR-15 and FR-16.	Privacy	High	v1
NFR-13	Access to personal data	At any moment, and in accordance with the GDPR articles 14, 15, 16 and 18, the SYSTEM should provide interested parties with access to their personal data, including the rights to rectify and remove it. This is related to FR-09, FR-10 and FR-11, as well as NFR-07.	Privacy	High	v1

2.3 Cokido

Cokido is a platform run by De Stuyverij, used in Belgium already. It provides a number of services and functionalities that have been presented in detail in Deliverable 1.2. In this section, we will describe the specific technologies, employed by Cokido, pointing out their key characteristics and discuss our choice not to utilize the current platform.

2.3.1 Front-end

Cokido's front-end was developed using AngularJS (v.1.7.2). AngularJS is a JavaScript-based front-end web application framework that aims to simplify both the development and testing of such as applications [1]. It provides us with a framework for client-side Model-View-Viewmodel (MVVM) and Model-View-Controller (MVC) architectures, including components, which are generally used in rich

Internet applications. The framework adapts and extends traditional HTML to present dynamic content through two-way data-binding that allows for the automatic synchronization of models and views.

2.3.2 Back-end

Cokido's back-end was based on combination of Laravel. Laravel [2] is a web framework that assists the development of web applications. It follows the MVC architecture and is based on Symfony. Laravel attempts to simplify and ease common tasks used in most web projects such as routing, authenticating and caching, by providing a unified syntax, along with all the necessary tools.

2.3.3 Assessment

With respect to the front-end, since the development of Cokido, AngularJS has been re-written from the ground up, providing new features and is now referred to as Angular (or Angular 2+) [2]. That has led many developers to migrate to the newer version, resulting in AngularJS' steady loss of community base and support. However, this migration comes at the cost of significant code refactoring effort.

Regarding the back-end, an alternative to Laravel, that many developers switch to, is NodeJS. NodeJS has some clear advantages over Laravel, as a Single Page App (SPA) backend. It enables the developer to use one language in the project (JavaScript), as opposed to using different languages for the backend and frontend development, which makes it easier to code. Furthermore, it creates the opportunity to share code between the frontend and backend and even make the app isomorphic. NodeJS has quicker page loading times, as it allows server-side rendering, which means that a page can be rendered at the server before it hits the browser.

All in all, the technologies employed in Cokido may have a number of advantages, but there are newer state-of-the-art technologies that provide us with additional features, that will enrich our platform and ensure its longer-term community support. In addition, taking into consideration the major code refactoring that needs to be done to utilize the current platform we decided to scrap the current platform and re-write it from scratch, as the effort of doing so is comparable with the one of getting accustomed with the current code and assuming its ownership, while the benefits of the former are clear.

2.4 Native or Web apps?

When developing a mobile app, one should decide based on the main objectives and overall goals whether to build a web app, native app or hybrid app. To do so, there are a variety of factors to take into consideration.

On the one hand, web apps are straightforward and quick to build, as well as easy to maintain. They are an inexpensive option and can run on all platforms, as long as the device can run a browser. In addition, a web app can update itself without the user's involvement. However, web apps lack the ability of native apps to leverage device utilities and sensors, such as sending push notifications, working offline and having access to location and camera. In addition, web apps are generally slower than native apps.

On the other hand, native apps, are very fast and responsive as they are platform-specific. They don't require an internet connection and allow the developer to access all the features of the chosen platform. Still, they come with some disadvantages; They are more expensive and not the best option when building a simple app. Native apps must be written in a specific and sometimes complex

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

language, that the chosen platform supports. Furthermore, in order to maintain a native app, the user needs to keep downloading updates.

Hybrid apps are built on web technology and are much easier to develop. They are cheaper than a native app and like web apps, they support all platforms. In addition, there is no need for a browser as opposed to a web app. Hybrid apps also have access to the device's internal APIs and can leverage its features. They are faster to developer than native apps because there is a single code base. However, they are slower and less interactive than native apps, while also being more expensive than web apps and dependent on a third-party platform.

In our case, a hybrid app is the best choice among the given options. It provides us with a speedy and easy to maintain build and considering that the specified application is not performance-intensive, further corroborates our choice. In addition, by taking advantage of technologies such as the Apache Cordova[4], we only need to build just a single app, which will work for all platforms. It also lets us leverage the device's utilities such as sending push notifications, which is an important feature based on the given functional requirements.

3 Families_Share platform technologies

3.1 MERN stack

MERN stack[5] is a free software programming package that is used for development of web applications. The main advantage of MERN is that JavaScript programming language is used both for the front-end and back-end part of the application. Specifically, MERN stands for MongoDB, Epxress.JS, React.JS and Node.JS. MongoDB, is used as the database of the stack, Node.JS with Express.JS is used for the back-end part and React.JS for the front-end part.

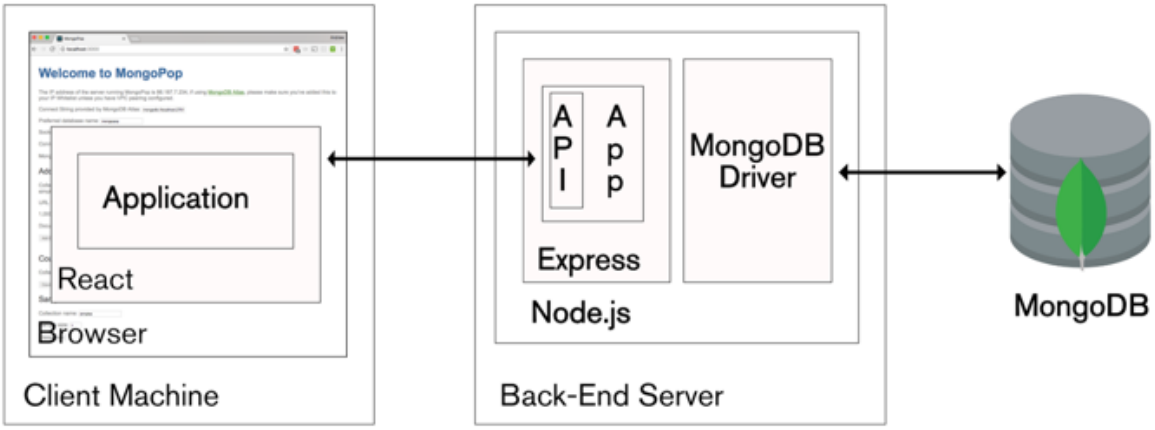

Figure 3: MERN stack architecture

It is worth to be noted that besides the MERN stack, there are also other software packages (stacks) that use JavaScript both on front-end and back-end such as the MEAN[1] Stack or the MEVN[2] Stack. The use of such stacks has increased exponentially the past few years and many experts suggest that soon the majority of new web apps will be programmed with those JavaScript stacks rather than the traditional web technologies which use the LAMP[3] stack.[6]

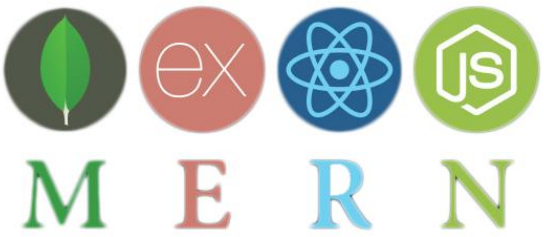

Figure 4: MERN stack logo

[1] MongoDB, Express.JS, Angular.JS, Node.JS

[2] MongoDB, Express.JS, Vue.JS, Node.JS

[3] Linux, Apache, MySQL, PHP/Python/Perl (PHP mostly)

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

3.1.1 MongoDB

MongoDB[7] is a noSQL database which uses documents in the form of JSON (JSON Documents). Specifically, the database is organized with collections, and in each collection there are JSON documents which are made up with field and value pairs, as depicted in Figure 5. In that particular example, *users* is the name of the collection while the document that it owns, includes the *name, age, status* of each user.

Figure 5: MongoDB document

Some of the **advantages** of using MongoDB are listed below:

- Large volumes of structured, semi-structured, and unstructured data
- Agile sprints, quick iteration, and frequent code pushes
- Object-oriented programming that is easy to use and flexible
- Efficient, scale-out architecture instead of expensive, monolithic architecture
- No over-complicated joins

3.1.2 React.JS

React.JS[8] (also known as React or ReactJS) 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. Complex React applications usually require the use of additional libraries for state management, routing, and interaction with an API.

Basic usage

In the code of [8] there is a rudimentary example of React usage in HTML with JSX and JavaScript.

The `Greeter` class is a React component that accepts a property `greeting`. The `ReactDOM.render` method creates an instance of the `Greeter` component, sets the greeting property to `'Hello World'` and inserts the rendered component as a child element to the DOM element with id `myReactApp`.

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

```
<div id="myReactApp"></div>

<script type="text/babel">
  class Greeter extends React.Component {
    render() {
      return <h1>{this.props.greeting}</h1>
    }
  }

  ReactDOM.render(<Greeter greeting="Hello World!" />, document.getElementById('myReactApp'));
</script>
```

Figure 6: React basic example

When displayed in a web browser the result will be the code in Figure 7.

```
<div id="myReactApp">
  <h1>Hello World!</h1>
</div>
```

Figure 7: React browser result

Notable features

Some notable features of React.JS are listed below:

- **One-way data binding with props**: Properties (commonly, props) are passed to a component from the parent component. Components receive props as a single set of immutable values (a JavaScript object).
- **Stateful components**: States hold values throughout the component and can be passed to child components through props:
- **Virtual DOM:** Another notable feature is the use of a "virtual Document Object Model", or "virtual DOM". React creates an in-memory data structure cache, computes the resulting differences, and then updates the browser's displayed DOM efficiently. This allows the programmer to write code as if the entire page is rendered on each change, while the React libraries only render subcomponents that actually change.
- **Lifecycle methods:** Lifecycle methods are hooks which allow execution of code at set points during the component's lifetime.
    - `shouldComponentUpdate` allows the developer to prevent unnecessary re-rendering of a component by returning false if a render is not required.
    - `componentDidMount` is called once the component has 'mounted' (the component has been created in the user interface, often by associating it with a DOM node). This is commonly used to trigger data loading from a remote source via an API.
    - `render` is the most important lifecycle method and the only required one in any component. It is usually called every time the component's state is updated, reflecting changes in the user interface.
- **JSX:** JSX (JavaScript eXtension) is an extension to the JavaScript language syntax. Similar in appearance to HTML, JSX provides a way to structure component rendering using syntax familiar to many developers. React components are typically written using JSX, although they do not have to be (components may also be written in pure JavaScript). JSX is similar to another extension syntax created by Facebook for PHP, XHP.

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

- **Nested elements**: Multiple elements on the same level need to be wrapped in a single container element such as the element shown above, or returned as an array.
- **Architecture beyond HTML:** The basic architecture of React applies beyond rendering HTML in the browser. For example, Facebook has dynamic charts that render to <canvas> tags and Netflix and PayPal use isomorphic loading to render identical HTML on both the server and client.

3.1.3 Node.JS & Express.JS

Node.js[9] is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code outside of a browser. Historically, JavaScript was used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser. Node.js lets developers use JavaScript to write 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 [10], unifying web application development around a single programming language, rather than different languages for server side and client side scripts.

Though .js is the conventional filename extension for JavaScript code, the name "Node.js" does not refer to a particular file in this context and is merely the name of the product. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

Platform architecture

Node.js brings event-driven programming to web servers, enabling development of fast web servers in JavaScript. Developers can create highly scalable servers without using threading, by using a simplified model of event-driven programming that uses callbacks to signal the completion of a task. Node.js connects the ease of a scripting language (JavaScript) with the power of Unix network programming.

Node.js was built on the Google V8 JavaScript engine since it was open-sourced under the BSD license. It is extremely fast and proficient with internet fundamentals such as HTTP, DNS, TCP. Also, JavaScript was a well-known language, making Node.js immediately accessible to the entire web development community.

Threading

Node.js operates on a single thread event loop, using non-blocking I/O calls, allowing it to support tens of thousands of concurrent connections without incurring the cost of thread context switching. The design of sharing a single thread among all the requests that use the observer pattern is intended for building highly concurrent applications, where any function performing I/O must use a callback. To

accommodate the single-threaded event loop, Node.js uses the *libuv* library—which, in turn, uses a fixed-sized thread pool that handles some of the non-blocking asynchronous I/O operations.

A downside of this single-threaded approach is that Node.js doesn't allow vertical scaling by increasing the number of CPU cores of the machine it is running on without using an additional module. However, developers can increase the default number of threads in the libuv thread pool.

Express.JS

Express[11] is a minimal and flexible 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 developer's disposal, creating a robust API with Express becomes quick and easy. Ultimately, Express provides a thin layer of fundamental web application features, without obscuring Node.js features.

3.2 AUTH Rapid Application Deployment Environment

AUTH Rapid Application Deployment Environment (ARADE) implements a Platform-as-a-Service (PaaS) infrastructure, which targets at offering to developers an environment, infrastructure and services that facilitate development, deployment and run of web applications. Towards this direction, the key design targets can be summarized in the following points:

- Provide a way to easily develop web applications minimizing the burden of configuring multiple different technology stacks.
- Provide one-click deployment capabilities.
- Provide access to a variety of services, which facilitate the applications' functional and non-functional requirements.

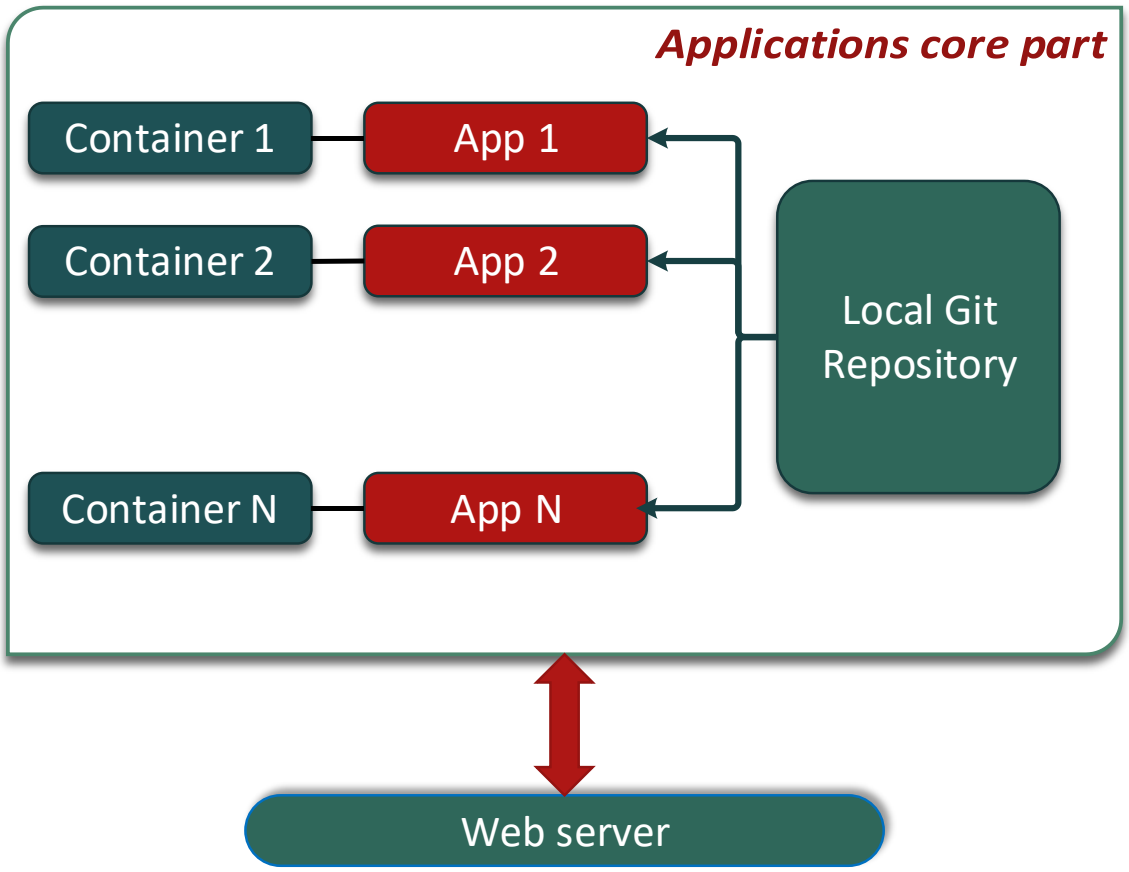

Figure 8: Deployment Environment Architecture

Figure 8 depicts the architecture of the Deployment Environment. Driven by the aforementioned key design points, the technical architecture has the following characteristics:

- Provision for **multiple isolated environments and services** interacting with each other through orchestration mechanisms provided by the system. This is enabled by the use of a Docker [12] infrastructure, providing a set of containers with different specifications that are created and started on-demand by the developers according to their needs.
- Incorporation of **Software as a Service components that provide RESTful APIs** [13], [14]. This is a fundamental decision that ensures extensibility and enables modular design, decoupling the services from the platform and making them accessible independently.

ARADE architecture is also based on the concept of having multiple isolated environments and services communicating with each other using the orchestration mechanisms provided by the same system. On top of that, it uses state-of-the-art approaches in order to provide a user-friendly interface combining flexibility and efficiency of use and at the same time robust correspondence with the advanced functional requirements that originate from software developers.

The key functionality of the Deployment Platform is the use of multiple containers created on demand by the platform users; each being composed of different pieces according to the developers' specifications. This enables all platform users to act independently without affecting the others and at

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

the same time allowing them to have access to all the necessary resources and services provided by the system.

As for the UI layer, ARADE enables its users to take advantage of the full functionality through their browser, making the platform accessible through almost every device that supports internet connection. Appropriate APIs are provided to facilitate the development of native applications.

Another characteristic is the fact that it incorporates Software-as-a-Service components accessible to all application containers through RESTful APIs. The decision of providing services through independent modules that comply with the REST design principles reduce coupling between the platform components and at the same time ensure extensibility and fault tolerance.

3.2.1 Key Components Overview

The key components that support the aforementioned architecture as illustrated in Figure 8 are the following:

1. ***Web Server***

The Web server lies between the platform users and the core applications' part enabling the interaction through a web-browser. The input of the Web server is a large series of events triggered by the users' browsers (clicks, inputs, etc...). Those events correspond to functionalities offered by the platform and involve the usage of the entire infrastructure. In addition, the Web server receives the response of every request performed into any part of the infrastructure so as to send the appropriate feedback to the platform users (e.g. Functionality performed successfully). The output of the Web server is two-fold as it; (i) performs the necessary requests to the backend components to trigger various functionalities and (ii) delivers the corresponding responses through the web browser.

2. ***Applications Core Part***

The Applications Core Part is a Docker-based infrastructure [15] responsible for hosting the applications of the developers registered in platform. Every application will be supported by a dedicated development environment (container) containing all the necessary development tools. The Applications Core Part is triggered by the web server and every other component that needs to perform operations on applications, containers and repositories. The Applications module returns the responses of the actions triggered to every respective module.

3. ***Local Git Repository***

The Deployment Platform will host a git-based code repository where developers will be able to store their applications' source code. This git-based code repository will also act as the authentication server. A different repository will be used for every application and each repository will be synchronized with the workspace of the associated application container. The local git repository will facilitate the users' management and will be triggered by the Applications Core Part in order to perform a series of operations (e.g. create/delete users and repositories). Full reference of the local git repository interface can be found at [16].

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

3.2.2 Implementation technologies

In an effort to support the previously described architecture and to ensure reliability and robustness, the implementation of the Deployment platform involves a set of state-of-the-art technologies. For the implementation of the PaaS part, the MEAN stack [17] has been used.

It is obvious from the above architecture and the key design points, that the Deployment Platform involves a series of low-level operations in order to manipulate and orchestrate the creation and usage of Docker containers towards developing and deploying web and mobile applications. In an effort to serve this type of functionality, on top of the aforementioned tools/frameworks, ***Python 3*** [18] has also been used. Python has been selected in order to take advantage of the expressiveness and the tools offered by a high-level programming language.

The implementation of the Deployment Platform involves a series of challenges that need to be faced in order to succeed in providing the functionalities identified at the requirements elicitation process. Those challenges along with the corresponding implementations will be presented in detail in the following paragraphs.

- ***Enable communication between Docker containers and host system***

The Deployment Platform functionalities require the collaboration between various Docker containers and the host system. Towards this direction, one of the major implementation challenges was to enable full communication between the host system and the Docker containers. As already noted, Docker containers are fully isolated environments that can be accessed only through the Docker-daemon, which acts as a communication channel. Although this channel enables sending commands in the container, the isolation principles do not allow the Docker-daemon to have any access to the container's inner procedures. It only provides the information regarding whether a command has successfully reached the container, but nothing regarding the outcome. This poses a major challenge in cases when full monitoring of the procedures performed inside a container is required. This is the case in the Deployment Platform, where the exact outcome of every procedure performed in each container is essential and affects the next action. In order for this to be even clearer let's consider the following application deployment scenario which consists of three steps:

1. **Synchronize container workspace with the application code repository.**
2. **Install code dependencies.**
3. **Deploy application.**

It is obvious that the platform needs to have information regarding the outcome of each of the three deployment steps in order to trigger different procedures based on the outcome. For example, in cases where synchronization fails (i.e. wrong credentials or the code repository is not reachable etc…), the platform has to notify the developer by providing a full log of the error and restrict the execution of the rest two steps. This functionality is not possible via the Docker daemon, which will respond that the "synchronize command" (mainly a git clone command) has successfully been sent to the container and nothing about the outcome.

In order to face this challenge, we investigated several possible solutions:

1. The first involves the generation of log files for every procedure performed inside every container. Upon completion of each process, those generated log file would be sent to the

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

host system via ssh command and stored into a dedicated location in order to become accessible. Although this solution works, it produces huge overhead and is both time and resources consuming especially in cases where there are many different active containers.

2. The second solution involves creating a logs server where the containers would be able to perform POST requests and automatically send the logs of all procedures. Although this solution overcomes all the above overhead restrictions, it has several disadvantages. At first, given the fact that each container is an isolated environment, performing a POST request requires enabling outgoing communication through a specific port. This would be the port exposed by the logs server. But this cannot be the case as all the outgoing traffic is being served by the Docker daemon and multiple containers cannot use the same outgoing port. In addition, even if this was faced by using a different dedicated container only for sending the logs to the logs server, (the communication between containers does have the aforementioned restrictions) the solution would result into loss of portability as a migration to a different logs server would work only for new containers and not for the pre-existing ones.

Based on the careful inspection of the aforementioned possible solutions and outcomes, we came up with a solution that enables the communication between the containers and the host system with the use of neither files, nor dedicated module (logs server).

The solution involves creating a mobile-age-library, common for all containers, which has a set of functions that both perform the required actions inside containers and at the same time enable the direct output of all the required output information to the Docker daemon. This process is performed by creating a PIPE channel using the INPUT channel created by the Docker daemon when sending the command. The channels STDERR and STDOUT are redirected in this channel and thus are directly accessible to the Docker daemon and as a result to the host system.

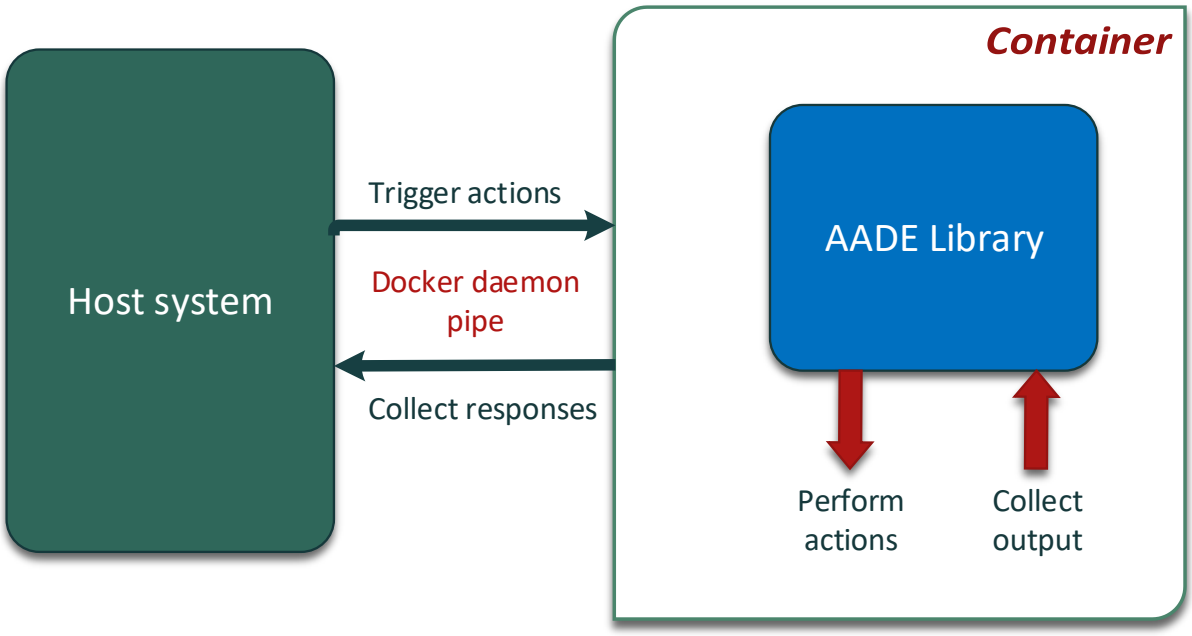

Figure 9: Communication diagram between the containers and the host system

Figure 9 illustrates how communication between the host system and the containers is enabled through the Mobile-Age library. This library has been implemented in Python.

- ***Implementation of the internal library***

As already noted, the need to run multiple procedures inside the Docker containers along with maintaining full communication with the host system requires the development of a dedicated library included inside every container. Given that one of the key design principles of the platform is extensibility, the challenge behind the implementation of this library is the following features:

1. **It should provide a central update mechanism**

The containers library includes all the functions that enable each of the procedures that run inside every container. To that end, due to the fact that those procedures are subject to change with the addition of new functionalities or with fixes on the already existing ones, the library should enable automatic update capabilities. Towards this direction, we designed a mechanism that enables the automatic update of the library when the system identifies that a new version has been released. In that way, the platform enables all the containers to be up-to-date.

2. **The host system should be able to call every function using the same endpoint**

Another feature towards extensibility is the ability of the host system to call via using the Docker daemon each one of the functions included in the library. Towards this direction, in order to avoid having to update the backend of the host system every time an update was made to the library, we created a mechanism able to dynamically run all functions included in the library.

The dynamic function calls functionality is performed by using function signatures, which provide a way to fully describe a function declaration. An example of a function signature is displayed in Figure 10, which corresponds to the definition: `def synchronize_repo(username, repo_url).`

```
{
    "function_name": "synchronize_repo",
    "params": {
        "username": "<the developer username>",
        "repo_url": "<the application repository url>"
    }
}
```

Figure 10: Function signature example

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

3.3 Development process

In order to effectively manage the software development process, we have adopted the JavaScript web application starter kit for the MERN (MongoDB, ExpressJS, ReactJS, NodeJS) stack, along with the software development lifecycle approach of Cyclopt [19] (js-starter-kit approach from now on). This includes a number of tools and recommendations for technical project management, code monitoring and evaluation as well as DevOps templates for Continuous Integration / Continuous Deployment (CI/CD).

The js-starter-kit approach includes the following ingredients:

- An agile project management approach for building specifically Software as a Service (SaaS) web (and mobile) applications.
- An approach to handle the codebase and to incorporate the GitFlow process into the daily grind of developers.
- How the different environment and the different servers should be set up
- What tools the developer should install and what tasks need to be automated
- An appropriate testing strategy to ensure functional suitability and a series of tools to implement it
- Continuous Integration and Continuous Development templates for various CI/CD systems
- Best practice security guidelines

To that end, a dedicated software repository has been set up on GitHub, along with a corresponding project. Tasks and requirements have been properly categorized into GitHub issues and the development process is organized in an agile manner, using a Kanban board approach.

Figure 11 depicts a screenshot from the Families_Share platform Project on GitHub.

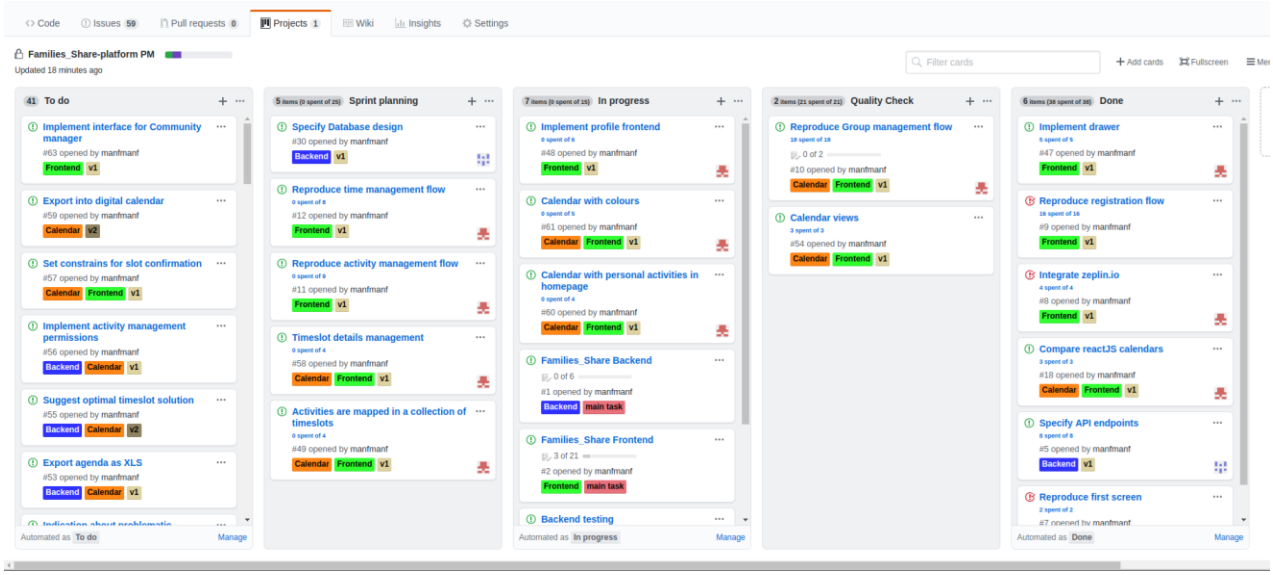

Figure 11: Kanban board for managing Families_Share platform development

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Monitoring and evaluating the progress of both the software project management and the source code is also a core part of any complete development process. To enable the provision of such services to the project developers and managers, Cyclopt will integrate a Quality as a Service (QaaS) module to the GitHub repository.

The QaaS module will be responsible for triggering analyses of the agile process and the source code at different timestamps and with different contents:

- *Per-commit analyses* will be triggered after the developer has pushed a commit to the repository and shall contain errors, warnings and suggestions with respect to the commit source code. The developer will be notified immediately, in-context and in-time based, in order to attend the issues while working on the problem at hand as soon as possible.
- *Weekly or monthly analyses* will be triggered automatically and will contain a full report on how the project is moving, an assessment report of the software quality of the project, broken down into quality characteristics (security, maintainability, functional suitability, etc.), and a series of recommendations on how to increase its quality. The report will be delivered to managers and developers to their channel of choice (email, pdf, slack etc.)

Of course, every project member can at any time log into the system and look at the latest analysis of the project.

Example aspects (overall quality and maintainability) of such a report in an be seen in Figure 12 and Figure 13 respectively.

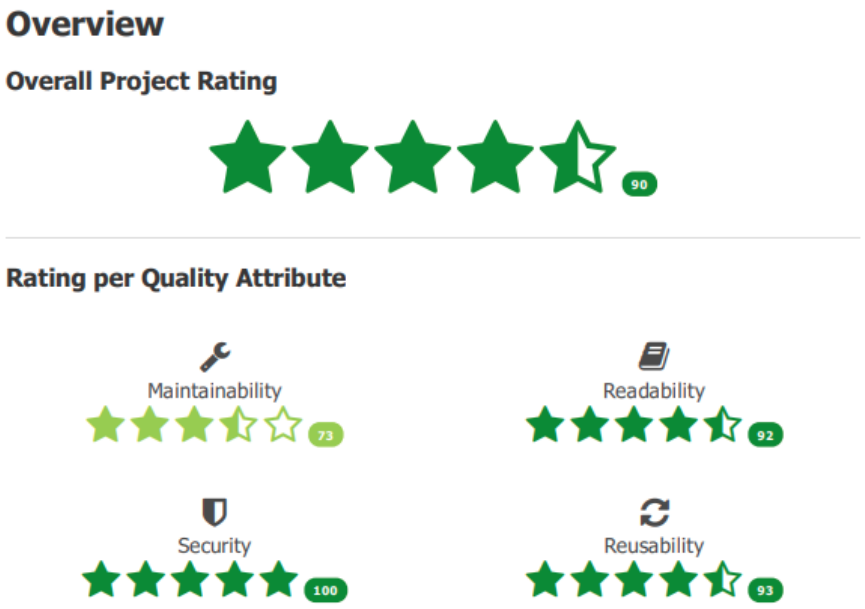

Figure 12: Overall software quality assessment example report

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

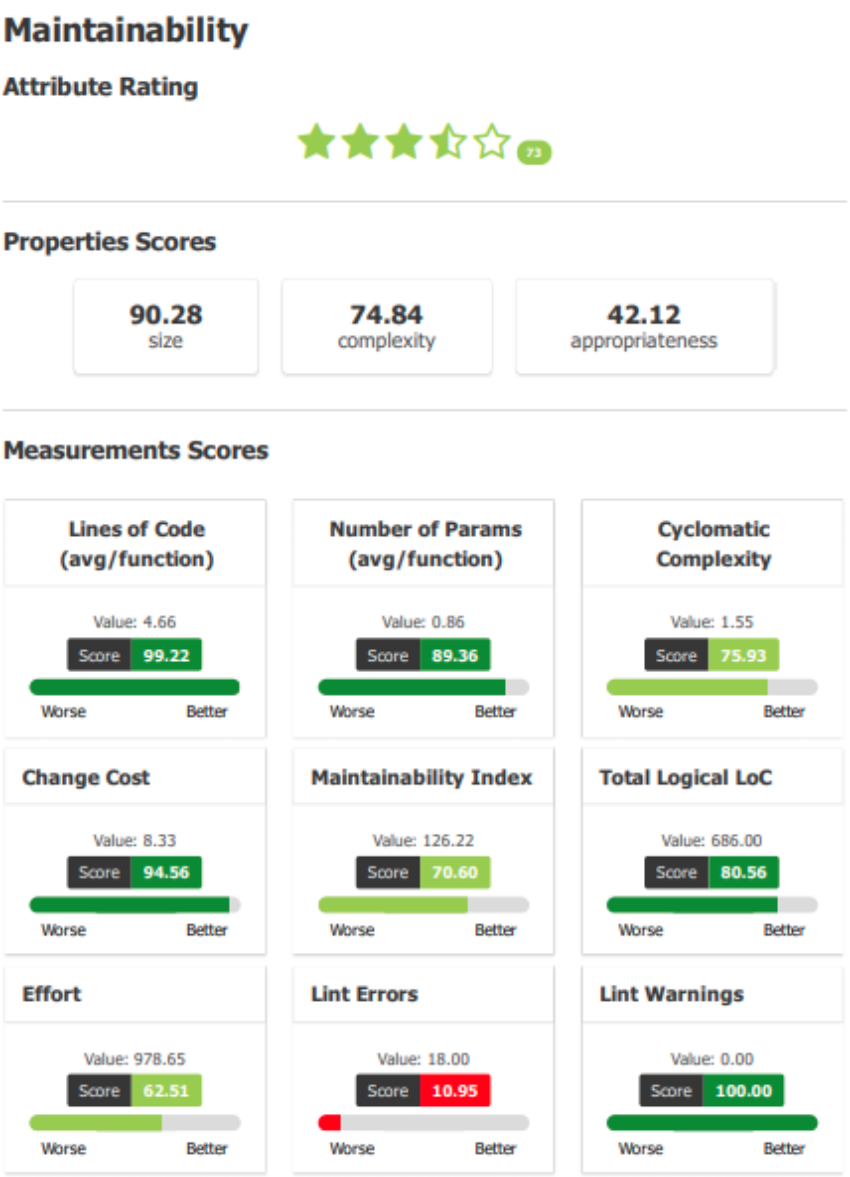

Figure 13: Report on the maintainability software quality characteristic

3.4 Calendar front-end components

As it becomes evident from the functional requirements presented in section 2, together with the app mockups specified in the Venice co-design sprint and included in Deliverable D1.2, the calendar is a fundamental functionality of the app. The calendar will enable users to plan, organize and conduct their groups, activities and timeslots and it is important to incorporate a corresponding front-end module into the apps that will be expressive, flexible and user-friendly.

In this section, we review and compare three calendar components that are compatible with React.JS.

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

3.4.1 FullCalendar

FullCalendar [20] is an open source event calendar based on JavaScript. It displays a full-size drag-n-drop calendar, which supports different views and offers a variety of customization capabilities. FullCalendar can be easily incorporated into an existing project via Node.js Package Manage (Npm).

Functionalities:

1. Views:
   FullCalendar supports at the moment the following views:
   - Month View
   - Agenda View
   - List View
   - Basic View
   - Custom Views

   All views are customizable. The user can modify existing titles and buttons as well as define new ones. Specific styles, regarding colours, fonts and sizes, can also be applied to the view in order to adjust its appearance.

2. Theme:
   The user has the ability to select a different theme for the entire calendar. A wide variety of free and premium themes can be found online.
3. Date Clicking and Selecting:
   The calendar detects when the user clicks on specific dates or times and provides him/her with the ability to select multiple dates or time slots with their mouse or touch device.
4. Events:
   FullCalendar can inherently display events from a single or multiple public Google Calendars. Events from private Google Calendars can be displayed utilizing FullCalendar's event hooks. However, it is up to the user to add this functionality. Furthermore, events' appearance (e.g. background colour and text colour) is also customizable.
5. Calendar locale and time zones:
   The user can tailor the calendar for certain languages (aka "locales"). The locale setting it the most important, as it sets the defaults of many other options at the same time.
6. Touch support:
   FullCalendar's Touch support includes smooth scrolling, time-range selection via long-press, and event drag-n-drop/resizing via long-press.

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Figure 14: FullCalendar in monthly view

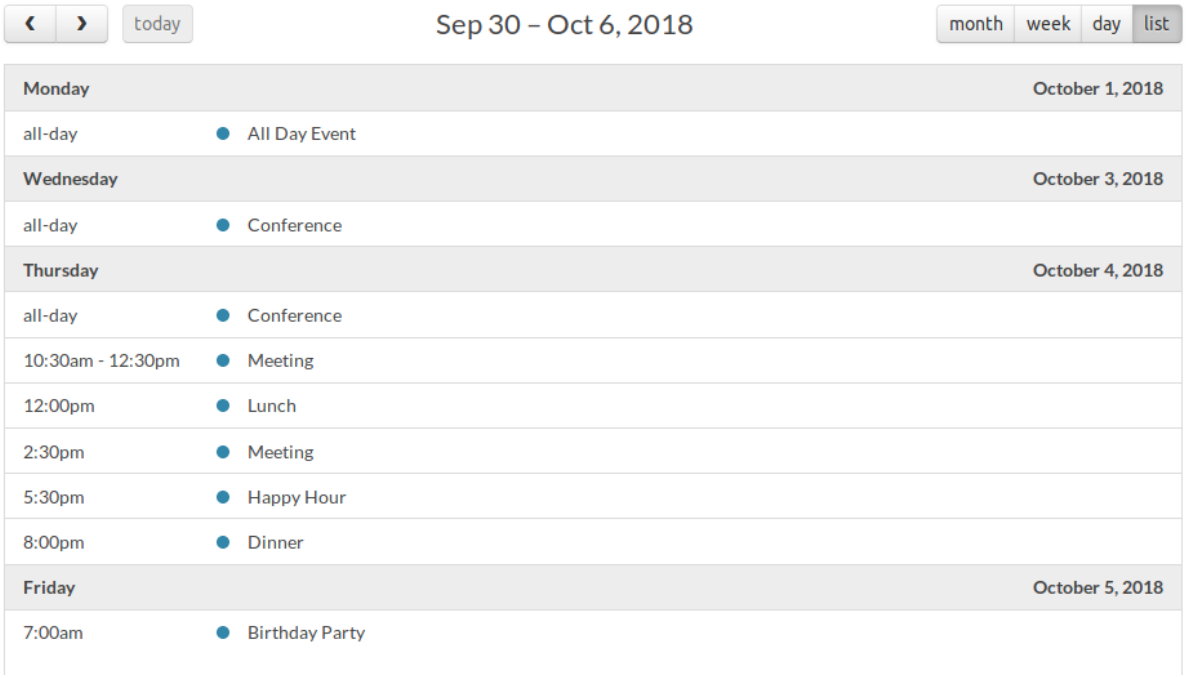

Figure 15: FullCalendar in list view

3.4.2 BigCalendar

BigCalendar [21] is an events calendar component built for React and made for modern browsers. It utilizes flexbox over the classic tables-ception approach. BigCalendar, inspired by FullCalendar, also offers different calendar views and a wide variety of customization capabilities.

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

1. Views:
   BigCalendar supports at the moment the following views:
   - Month View
   - Week View
   - Work Week View
   - Day View
   - Agenda View
   - Custom Views

   The user can select which views are available for display. Furthermore, Views' appearance is customizable as the user can set the height of the content and select specific colours for different areas of the calendar etc. BigCalendar provides the user with the ability to create and render custom components, thus modifying not only the Views' appearance but also

2. Date Clicking and Selecting: BigCalendar not only catches the user's clicks on specific dates and timeslots but allows him/her to select multiple dates and timeslots.
3. Events: It is up to the user to provide the calendar with the list of events to be displayed. By default BigCalendar Events have certain properties, such as title, start time etc. However, the user may provide a custom Event component to the calendar, in order to extend the default properties by adding new ones and customize the appearance of the events (e.g. the user can provide explicit styles for different event types).
4. Calendar locale and time zones:
   There is no feature at the moment that provides this capability. Instead the user can use one of the many excellent solutions already out there, via adapters called "localizers". Big Calendar comes with two localizers for use with Globalize.js or Moment.js.
5. Touch support:
   BigCalendar supports drag and drop functionality, thus allowing the user to freely select and move events on the calendar.

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

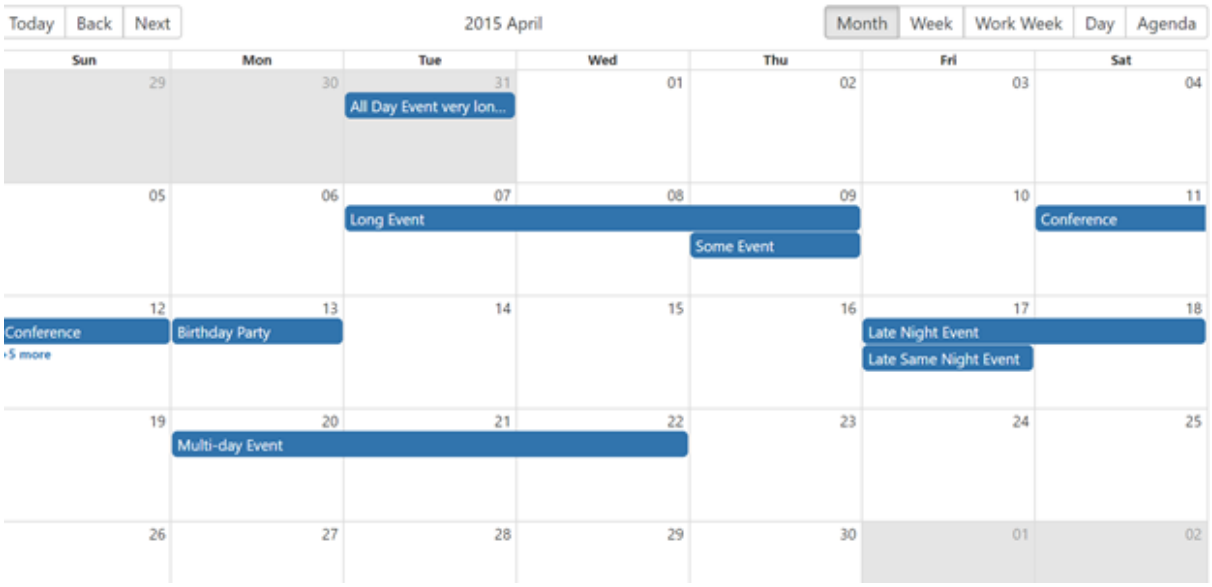

Figure 16: BigCalendar in monthly view

Today | Back | Next | 4/1/2015 — 5/1/2015 | Month | Week | Work Week | Day | Agenda

Date	Time	Event
Tue Apr 07	12:00 am »	Long Event
Wed Apr 08	« all day »	Long Event
Thu Apr 09	« all day »	Long Event
	12:00 am »	Some Event
Sat Apr 11	12:00 am »	Conference
Sun Apr 12	« all day »	Conference
	10:30 am — 12:30 pm	Meeting
	12:00 pm — 1:00 pm	Lunch
	2:00 pm — 3:00 pm	Meeting
	5:00 pm — 5:30 pm	Happy Hour
	8:00 pm — 9:00 pm	Dinner
Mon Apr 13	7:00 am — 10:30 am	Birthday Party
Fri Apr 17	7:30 pm »	Late Night Event
	7:30 pm — 11:30 pm	Late Same Night Event
Sat Apr 18	« 2:00 am	Late Night Event
Mon Apr 20	7:30 pm »	Multi-day Event
Tue Apr 21	« all day »	Multi-day Event
Wed Apr 22	« 2:00 am	Multi-day Event

Figure 17: BigCalendar in list view

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

3.4.3 Ant Design Calendar

Ant Design [22] is a React UI library that contains a set of high quality components and demos for building rich, interactive user interfaces. This library can be installed via Node.js Packet Manager (Npm). Ant Design Calendar [23] is an out of the box component provided by this library, that acts a container for displaying data in calendar form.

1. Views:

Ant Design Calendar offers only two different views at the moment:

- Month View
- Year View

The Views' appearance is customizable as the user can provide different styles for the display of the month cell.

2. Events:

Ant Design Calendar supports three types of Events:

- Warning Event
- Usual Event
- Error Event

Unfortunately, Ant Design Calendar offers no customization capability regarding the events.

3. Calendar locale and time zones:
   There is no feature at the moment that provides this capability. Instead, the user can use Node's Packet Manager to import an external localizer such as Moment.js.

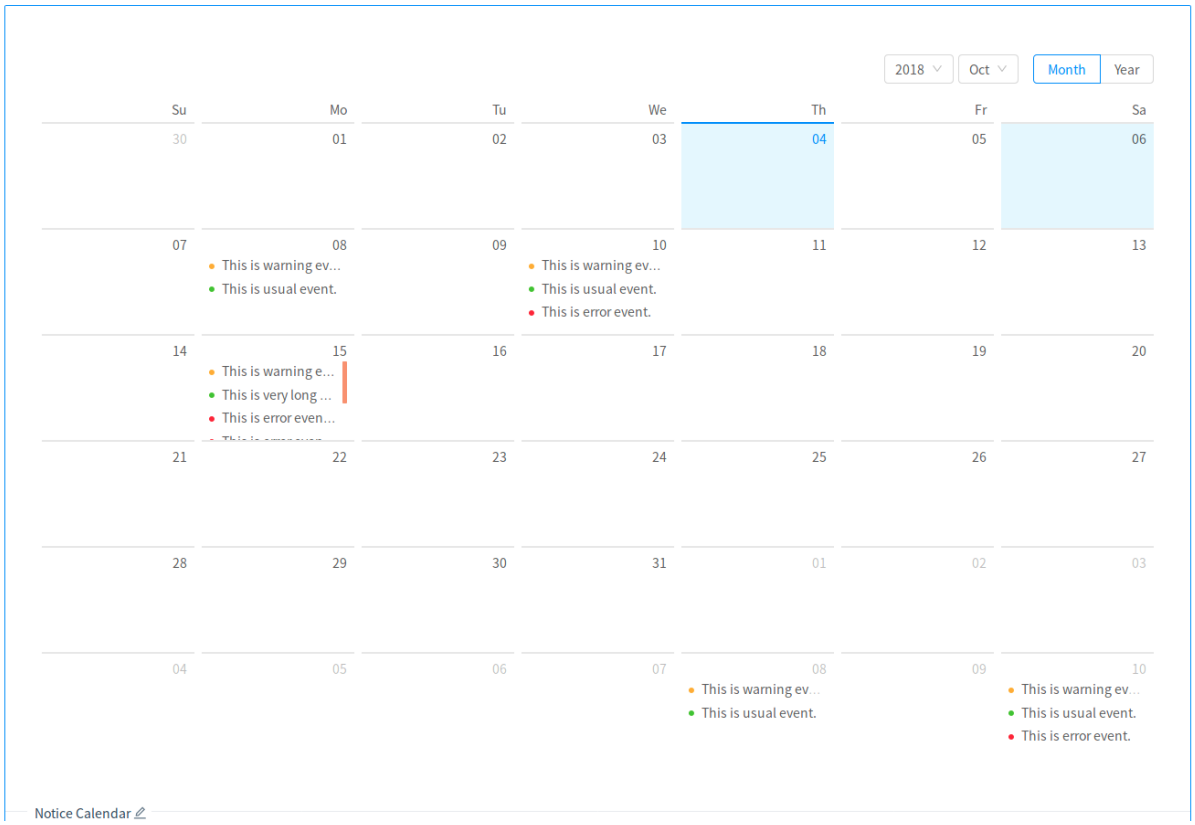

Figure 18: Ant Design Calendar in month view

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

3.4.4 Comparison of key functionalities

Table 3 summarizes the key functionalities of the aforementioned React.JS calendar components.

Table 3: Comparison of functionalities of calendar components

	FullCalendar	BigCalendar	Ant Design Calendar
Month View	✓	✓	✓
List View	✓	✓	
Customizable View Appearance	✓	✓	✓
Customizable Event Appearance	✓	✓	
Language Locale	✓	✓ *	
Time Locale	✓	✓ *	✓
Display Events	✓	✓	✓
Custom View and Event Components		✓	
Date Clicking and Selecting	✓	✓	✓

* supported via external package

From the previous, we are tentatively selecting BigCalendar, as it provides a customizable event component, which is necessary in order to comply with the requirement *FR57: Timeslot information* and reproduce the corresponding mock-ups.

An example of a custom event component is presented in Figure 19, that displays a timeslot that takes place two days, and involves labels and numbers. These labels can easily be changed and customized.

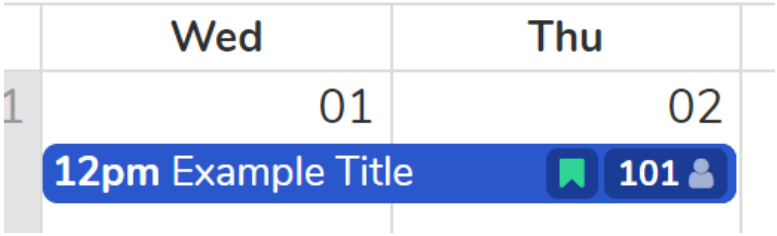

Figure 19: Custom Event Component example, using BigCalendar.

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

4 Families_Share platform specification

4.1 Software Architecture

The Families_Share software architecture will be based on the generic MERN stack architecture, as discussed in section 3.1. However, in order to provide seamless integration with the user's mobile device, a corresponding native app will be developed. Based on WebView technologies, the apps will first be available for Android devices in the platform version 1, while iOS is planned to be supported in version 2.

Consequently, the Families_Share apps will be *Hybrid* i.e., while being device agnostic and based on web technologies, they will be able to use the native API of the device and therefore have access to its various sensors and resources, such as camera, location and calendar. At the same time, as the front-end design will be responsive, one will be able to access the platform from their desktop device using a web browser.

The platform architecture is illustrated in Figure 20.

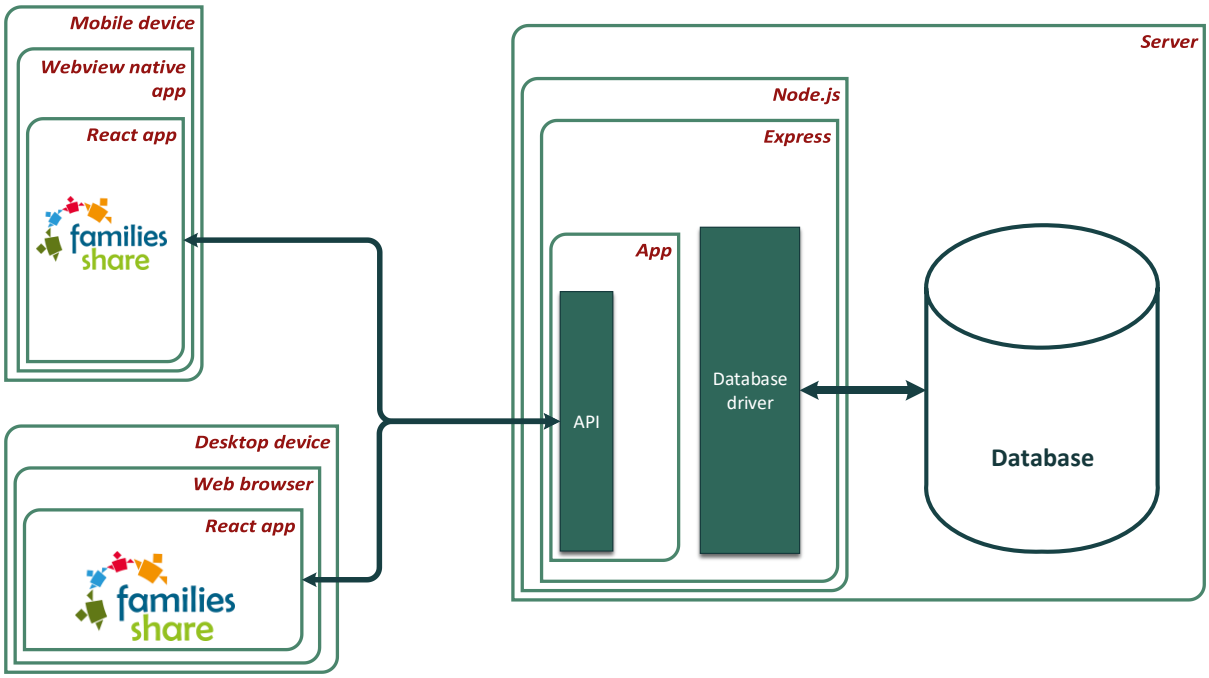

Figure 20: The Families_Share platform software architecture

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

4.2 Deployment Plan

For the deployment of the Families_Share platform, we will employ the AUTH Automatic Deployment Environment, as described in section AUTH Rapid Application Deployment Environment3.2.
The deployment is illustrated in Figure 21 and will be based on a Dockerized approach, according to which, each app will run within its own separate container. That means that apps will be independent of one another, having the ability to start, stop and be updated and possibly adjusted with different programming libraries and/or development environments in an individual manner and without affecting the rest. Moreover, the containers will include the respective databases, yielding the latter inaccessible to external apps. Finally, the apps will have the ability to make use common services, which will be provided in a centralized fashion in a Software-as-a-Service manner, in order to exploit their potential commonalities.
What is more, the ability of automatic deployment can enable rapid prototyping, which can in turn, greatly facilitate co-design workshop activities. Workshop participants can immediately see how their ideas and suggestions affect the end result and have the ability to provide immediate feedback and further adjustments.

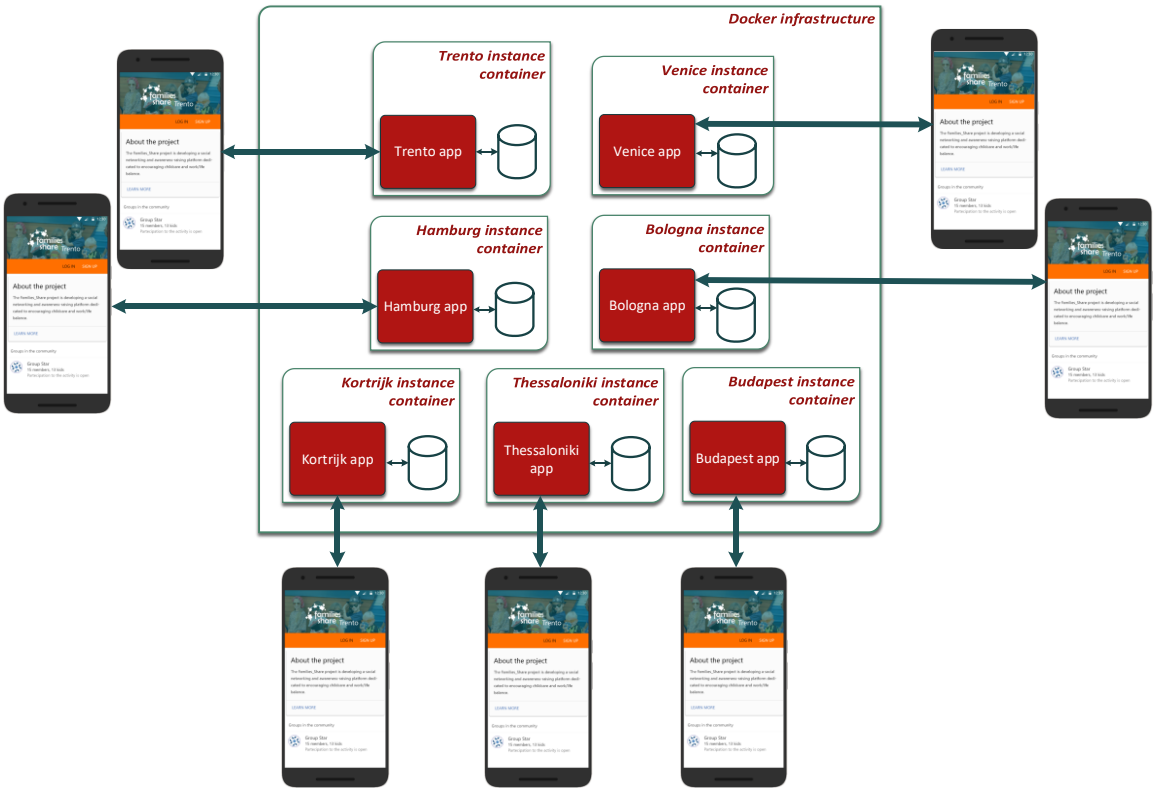

Figure 21: The Families_Share platform deployment architecture

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

4.3 Platform API

In order to provide its services to the front-end of each app, the platform backend can be accessed using an API. In this section, we provide a list of endpoints that will constitute the back-end API, specified using a RESTful approach. These endpoints grant access to the data stored in the back-end and have been designed to facilitate the work carried out in the front-end.

The following tables present the endpoint id, its relative URL, its description, the method(s) it supports and, optionally, notes on its functionality and use.

Id: Endpoint	`EP-01: /addresses/{id}`
Method	GET, PUT
Description	Show or update a single address
Notes	

Id: Endpoint	`EP-02: /cities`
Method	GET
Description	List all cities, this is used to link a city to profiles and groups.
Notes	

Id: Endpoint	`EP-03: /cities/{id}`
Method	GET
Description	Show a specific city
Notes	

Id: Endpoint	`EP-04: /groups`
Method	GET
Description	Get a list of all groups
Notes	

Id: Endpoint	`EP-05: /groups`
Method	GET
Description	Search for a group by name, postcode or city name
Notes	

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Id: Endpoint	EP-06: /groups
Method	POST
Description	Create a new group, where the current user is an administrator.
Notes	Including group settings as parameters

Id: Endpoint	EP-07: /groups/{id}
Method	GET, PUT, DELETE
Description	Show, update or delete a single group
Notes	

Id: Endpoint	EP-08: /groups/{id}/restore
Method	POST
Description	Restore a group from its deleted state where the current user is the owner
Notes	

Id: Endpoint	EP-09: /groups/{id}/invites
Method	GET
Description	Show the invites for a specific group.
Notes	

Id: Endpoint	EP-10: /groups/{id}/invites
Method	POST
Description	Create a group invite for a group where the current user is an administrator
Notes	Email, or phone number as parameters

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Id: Endpoint	EP-11: /groups/{id}/invites/{id}
Method	GET, DELETE
Description	Get or cancel a single group invite for a group where the current user is an administrator.
Notes	

Id: Endpoint	EP-12: /groups/{id}/announcements
Method	GET
Description	List all announcements posted in a group where the current user is an accepted member
Notes	

Id: Endpoint	EP-13: /groups/{id}/announcements/{id}
Method	GET
Description	Get a specific group announcement posted in a group where the current user is an accepted member
Notes	

Id: Endpoint	EP-14: /groups/{id}/announcements
Method	POST
Description	Post an announcement in a group where the current user is an accepted member
Notes	-Title -Body -Image

Id: Endpoint	EP-15: /groups/{id}/announcements/{id}
Method	PUT, DELETE
Description	Update or delete a specific group announcement posted in a group where the current user is an administrator (or message author)
Notes	

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Id: Endpoint	EP-16: /groups/{id}/settings
Method	GET, PUT
Description	Get or update the settings of a group
Notes	- Visible or not visible - open or invite-only -means for external communication (WhatsApp/ Facebook links etc)

Id: Endpoint	EP-17: /groups/{id}/activities
Method	GET
Description	List all activities for a given group where the user is currently a member of
Notes	

Id: Endpoint	EP-18: /groups/{id}/activities/{id}
Method	GET
Description	Get a specific activity for a given group where the current user is an accepted member of
Notes	

Id: Endpoint	EP-19: /groups/{id}/activities
Method	POST
Description	Create (or propose) an activity for a given group where the current user participates
Notes	activity states/properties: -proposed -fixed -cancelled -completed Also: -Title, location, cost, image, description

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Id: Endpoint	EP-20: /groups/{id}/activities/{id}
Method	PUT, DELETE
Description	Update (modify/confirm/close) or delete a specific activity for a given group where the current user is an admin
Notes	

Id: Endpoint	EP-21: /groups/{id}/activities/{id}/timeslots
Method	GET
Description	Get all timeslots for a specific activity
Notes	

Id: Endpoint	EP-22: /groups/{id}/activities/{id}/timeslots
Method	POST
Description	Add a timeslot to an activity
Notes	

Id: Endpoint	EP-23: /groups/{id}/activities/{id}/timeslots/{id}
Method	GET, PUT, DELETE
Description	Get, update or delete a specific timeslot
Notes	

Id: Endpoint	EP-24: /groups/{id}/activities/{id}/timeslots/{id}/participants
Method	GET, PUT
Description	Get, update all participants for a time slot
Notes	

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Id: Endpoint	EP-25: /groups/{id}/activities/{id}/timeslots/{id}/participants/
Method	POST
Description	Set the participants of a timeslot
Notes	Users[], Children[] (might want to add specific endpoints to add self to participant list)

Id: Endpoint	EP-26: /languages
Method	POST
Description	Sets the language for the current user session
Notes	Might use local storage for this

Id: Endpoint	EP-27: /users
Method	GET
Description	Get a list of users that allow their profile to be listed.
Notes	

Id: Endpoint	EP-28: /users/{id}
Method	GET
Description	Get a single user
Notes	

Id: Endpoint	EP-29: /users/{id}/children
Method	GET
Description	Get a list of children of a user that allow their profiles to be listed.
Notes	

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Id: Endpoint	EP-30: /users/{id}/children
Method	POST
Description	Create a new child object for the current user
Notes	All child information should be passed as parameters here

Id: Endpoint	EP-31: /users/{id}/children/{id}
Method	GET, PUT, DELETE
Description	See, update or delete a single child object
Notes	To be used for exporting PDF children profiles

Id: Endpoint	EP-32: /users/{id}/children/{id}/parents
Method	GET
Description	List the parents of a child
Notes	

Id: Endpoint	EP-33: /users/{id}/children/{id}/parents
Method	POST
Description	Add another user as parent of a child
Notes	

Id: Endpoint	EP-34: /users/{id}/groups
Method	GET
Description	List the groups that a user belongs to
Notes	

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Id: Endpoint	EP-35: /users/{id}/groups
Method	POST
Description	Apply to join a group as the current user
Notes	

Id: Endpoint	EP-36: /users/{id}/groups/{id}
Method	DELETE
Description	Leave a group as the current user
Notes	

Id: Endpoint	EP-37: /users/{id}/invites
Method	GET
Description	List all invites created by the current user
Notes	

Id: Endpoint	EP-38: /users/{id}/invites/{id}
Method	GET, DELETE
Description	Get or cancel a specific invite as the current user.
Notes	

Id: Endpoint	EP-39: /users/{id}/invites
Method	POST
Description	
Notes	Email, phone number, inviteType (group, community, framily, parent of child)

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Id: Endpoint	EP-40: /users/{id}/export
Method	GET
Description	Export all information stored in the platform for a specific user, including profile, logs, activities and any other analytics
Notes	

Id: Endpoint	EP-41: /profiles
Method	GET
Description	List all profiles
Notes	

Id: Endpoint	EP-42: /profiles
Method	POST
Description	Create a profile
Notes	given_name, family_name, email, phone, facebook, picture, about, show_in_public, active_or_suspended, show_actual_name

Id: Endpoint	EP-43: /profiles/{id}
Method	PUT, DELETE
Description	Update or delete profile.
Notes	Update includes profile suspension

Id: Endpoint	EP-44: /notifications
Method	POST
Description	Create a new notification
Notes	Example parameters: “new user entered group”, “a new activity has been created”, “the agenda for a specific activity has changed” “phone” or “email”

This project has received funding from the European Union’s Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

Id: Endpoint	EP-45: /notifications
Method	GET
Description	Get all notifications related to the specific user
Notes	

Id: Endpoint	EP-46: /notifications/{id}
Method	GET, PUT
Description	Get or update a specific notification
Notes	

Id: Endpoint	EP-47: /community/settings
Method	GET, PUT, POST
Description	See, update or delete the community settings
Notes	- Open or closed community

Id: Endpoint	EP-48: /users/{id}/activities
Method	GET
Description	Get a list of activities for a specific user
Notes	

Id: Endpoint	EP-49: /groups/{id}/invites
Method	GET
Description	Get a list of activities for a specific group
Notes	

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

4.4 Satisfaction of requirements

ID: Title	*FR-01: Website for each community*		
Description	Each community has a website as companion of the app (each CityLab can have more than one community). Perhaps we could use second level domains for these (e.g. trento.families-share.eu).		
Requirement type	Functional	**Category**	Website
Priority	High	**Prototype Version**	Version 1
Fulfilled by	Frontend		

ID: Title	*FR-02: Website shows the list of groups for that community (as in Cokido*		
Description	We can create an endpoint within the app that returns information about public groups. So the website could be totally static, apart from the content shown from that endpoint. That way one could browse some public groups and perhaps see photos, just like in Cokido. If they click on the group however, they will need to download the app and register.		
Requirement type	Functional	**Category**	Website
Priority	Low	**Prototype Version**	Version 1
Fulfilled by	EP-4, Frontend		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-03: The app has different instances*		
Description	Each community will have its app to download from the Google Play Store (android first and then iOS later on).		
Requirement type	Functional	**Category**	General
Priority	High	**Prototype Version**	Version 1
Fulfilled by	The selected deployment architecture		

ID: Title	*FR-04: App design is mobile first*		
Description	The interface will be designed mainly for mobile phones but it can be accessed also from the web (for iOS and web users).		
Requirement type	Functional	**Category**	General
Priority	High	**Prototype Version**	Version 1
Fulfilled by	Frontend		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-05: Search group*		
Description	Users can search for groups from the web page or the mobile app.		
Requirement type	Functional	**Category**	Search
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-5		

ID: Title	*FR-06: Search users*		
Description	New users should be able to search for other users and ask them to join her/his group. Should it be optional?		
Requirement type	Functional	**Category**	Search
Priority	Low	**Prototype Version**	Version 2
Fulfilled by	EP-27		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-07: User name visibility*		
Description	Users should be able to decide if their name should be searchable (visible / not visible).		
Requirement type	Functional	**Category**	User profile Privacy
Priority	High	**Prototype Version**	Version 1
Fulfilled by	Frontend, EP-42, EP-43		

ID: Title	*FR-08: User search visibility*		
Description	Some users could decide to be invisible to the community. If they are not visible, to add them to a group or as a parent you should input the email or telephone number.		
Requirement type	Functional	**Category**	User profile Privacy
Priority	High	**Prototype Version**	Version 1
Fulfilled by	Frontend, EP-42, EP-43		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-09: Cancel profile*		
Description	Users should be able to cancel all their information.		
Requirement type	Functional	**Category**	User profile Privacy
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-43		

ID: Title	*FR-10: Suspend profile*		
Description	Users should be able to suspend their profile.		
Requirement type	Functional	**Category**	User profile Privacy
Priority	Medium	**Prototype Version**	Version 2
Fulfilled by	EP-43		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-11: Export all information*		
Description	Users should be able to export all their data stored in the platform (as JSON).		
Requirement type	Functional	**Category**	User profile Privacy
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-40		

ID: Title	*FR-12: Integrate children information within the platform*		
Description	Children information should be provided within the app (not in a printed form). Data should be encrypted.		
Requirement type	Functional	**Category**	Children info Privacy
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-30, EP-31, EP-32, EP-33		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-13: Showing children basic information + special needs*		
Description	Children basic information such as name, age and gender should be visible to the other members of the group (when organizing an activity). If there are any special needs, a symbol can be placed next to the children name. Integrate child card inside the app. So that we can search e.g. based on the age of the participants of the group. Age and gender are most important info too. Perhaps also add information about possible health issues (call it "special needs").		
Requirement type	Functional	**Category**	Children info
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	Frontend, EP-30, EP-31, EP-32, EP-33		

ID: Title	*FR-14: Managing who can view your full children information*		
Description	When users provide their children information, they should be notified and accept that this information can be shared with other members of the group. Users should be able to decide who can see this info. [Information sharing option]		
Requirement type	Functional	**Category**	Children info Privacy
Priority	High	**Prototype Version**	Version 2
Fulfilled by	Frontend, EP-31, further defined in V2		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-15: Export children information*		
Description	Users should be able to export in a PDF file the children information of the activity they are involved		
Requirement type	Functional	**Category**	Children info Privacy
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-31		

ID: Title	*FR-16: Privacy consent at the registration*		
Description	When users register in the platform, the platform should make sure that consent is "informed consent" and provide necessary privacy notice and other information at the time user consent is required. The registration process stops if user does not provide the consent.		
Requirement type	Functional	**Category**	Log-in / Registration Privacy
Priority	High	**Prototype Version**	Version 1
Fulfilled by	Frontend		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-17: Parents can add other users as parents*		
Description	Parents have different profile and can share the same children profile. A parent can be added as co-parent / guardian / family member.		
Requirement type	Functional	**Category**	Children info User profile
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	EP-33		

ID: Title	*FR-18: User hierarchy*		
Description	Community manager <- Group administrator <- User (parents / professionals) <- External [i.e. person not registered in the platform]. This means that the community manager is inherently a Group administrator for all groups. The Professional is a user role that has certain permissions.		
Requirement type	Functional	**Category**	User categories
Priority	High	**Prototype Version**	Version 1
Fulfilled by	Database design		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-19: New member notification*		
Description	When a new member enters a group all other members are notified.		
Requirement type	Functional	**Category**	Notification
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	EP-44		

ID: Title	*FR-20: New member acceptance rule*		
Description	Two options are provided: **Strict**: all members (who are also group administrators) they have to agree to the new member. **Flexible**: at least one group administrator agrees .		
Requirement type	Functional	**Category**	Group management Community management
Priority	High	**Prototype Version**	Version 2
Fulfilled by	EP-47		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-21: Single sign on*		
Description	This is an optional feature. We will start with registering for a local account functionality and perhaps integrate that in the future.		
Requirement type	Functional	**Category**	Log-in / Registration
Priority	Low	**Prototype Version**	Version 2
Fulfilled by	Frontend, OAuth		

ID: Title	*FR-22: Community default options*		
Description	Options when you create a community: • New members are automatically administrator OR not • Approval for new members: Do all group administrators need to agree? Or just one is fine? (i.e. AND or OR). [see #20] Group can change this option or not.		
Requirement type	Functional	**Category**	Community management
Priority	Medium	**Prototype Version**	Version 2
Fulfilled by	EP-47		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-23: Community access option*		
Description	The community can be open or not.		
Requirement type	Functional	**Category**	Community management
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	EP-47		

ID: Title	*FR-24: Group settings when creating a group*		
Description	A group can be visible or not visible: that is a group can be searchable or not. If it is visible, an external user can contact the group administrator for joining the group.		
Requirement type	Functional	**Category**	Group management Privacy
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	EP-06		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-25: Join a group with an invite*		
Description	All groups are invite-only. Some are visible and some are not. If it is not visible you have to receive an invite by the group administrator to join in. If the group is visible, you can contact the group administrator to receive an invite.		
Requirement type	Functional	**Category**	Group management
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-37, EP-38, EP-39		

ID: Title	*FR-26: Type of notifications*		
Description	Members and group administrators should receive notifications. System event notifications e.g: - “a new user entered the group” - “a new activity has been created’ - “the agenda for a specific activity was closed” The notification can be sent in the phone or by email.		
Requirement type	Functional	**Category**	Notifications
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-44		

This project has received funding from the European Union’s Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-27: Phone notifications*		
Description	Phone notifications are enabled.		
Requirement type	Functional	**Category**	Notifications
Priority	Medium	**Prototype Version**	Version 2
Fulfilled by	EP-44, hybrid app architecture		

ID: Title	*FR-28: Professional users*		
Description	Professionals can be invited by group administrators. They are users and can access information related to the activity in which they are involved. OR we can keep as it is in Cokido (information provided in the activity by the group administrator.		
Requirement type	Functional	**Category**	User categories
Priority	Medium	**Prototype Version**	Version 2
Fulfilled by	To be specified in V2		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-29: Assign group roles*		
Description	Group administrators can assign different roles to the members (e.g. location manager, financial manager). For now, it is just a text field. Possibly implements checklists (one for a "group administrator") for supporting in managing group roles.		
Requirement type	Functional	**Category**	User categories
Priority	Low	**Prototype Version**	Version 2
Fulfilled by	To be specified in V2		

ID: Title	*FR-30: Types of activities*		
Description	The interface should support activities that have two characteristics: - Recurrent vs One-time - Flexible vs Fixed User should not categorize the type of activity, this can be automatically inferred by the system.		
Requirement type	Functional	**Category**	Activity management
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-19, Frontend		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-31: Promote groups and activity*		
Description	We should provide indications or material for supporting the promotion of groups or activities (flyer, social media material, etc.).		
Requirement type	Functional	**Category**	Group management Activity management
Priority	Low	**Prototype Version**	Version 1
Fulfilled by	EP-18		

ID: Title	*FR-32: Evaluating an activity*		
Description	When closed, activities can be evaluated by their members (e.g. with a questionnaire, a summary of the activity, a checklist).		
Requirement type	Functional	**Category**	Activity management
Priority	Medium	**Prototype Version**	Version 2
Fulfilled by	To be specified in V2, Frontend		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-33: Sharing knowledge*		
Description	Once finished, information on activities and groups can be shared within the community through the website.		
Requirement type	Functional	**Category**	Activity management
Priority	Low	**Prototype Version**	Version 2
Fulfilled by	To be specified in V2		

ID: Title	*FR-34: Starting page / Homepage for new users*		
Description	Information displayed to the users when they access the platform: - Login Registration form - Info on Families_Shares project Info on active groups.		
Requirement type	Functional	**Category**	Interface
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	Frontend		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-35: Starting page / Homepage for registered users*		
Description	Information displayed to the registered users when they enter the application: - Calendar - Joined groups Notifications		
Requirement type	Functional	**Category**	Interface
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	Frontend, EP-05, EP-45, EP-17		

ID: Title	*FR-36: Provide checklists*		
Description	Provide users with checklists for supporting some activities (WhatsApp group creation, financial rules, etc.).		
Requirement type	Functional	**Category**	Interface
Priority	High	**Prototype Version**	Version 1
Fulfilled by	Frontend		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-37: Log-in with email*		
Description	Allows log-in with email address.		
Requirement type	Functional	**Category**	Log-in / Registration
Priority	High	**Prototype Version**	Version 1
Fulfilled by	Frontend, EP-41		

ID: Title	*FR-38: Log-in with SMS*		
Description	Allows log-in with SMS (not only with an email).		
Requirement type	Functional	**Category**	Log-in / Registration
Priority	Medium	**Prototype Version**	Version 2
Fulfilled by	To be specified in V2		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-39: Support communication within and outside the app*		
Description	Communication can be supported within the app (announcements / message) or outside (WhatsApp/Telegram/facebook). This can be decided by the community manager.		
Requirement type	Functional	**Category**	Communication Community management
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	EP-47, EP-14		

ID: Title	*FR-40: Announcements (a.k.a. Messaging) feature*		
Description	Kept as they are in Cokido (basic) but call them Announcements and have it for informing group members. This can also include info for support the group to manage a WhatsApp / telegram group. Users should be able to post a message and give replies.		
Requirement type	Functional	**Category**	Communication
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-12. EP-13, EP-14		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-41: Edit announcements / replies*		
Description	You can edit announcements or replies (created by you).		
Requirement type	Functional	**Category**	Communication
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	EP-15		

ID: Title	*FR-42: Deleting announcements / replies*		
Description	You can delete announcements or replies (created by you).		
Requirement type	Functional	**Category**	Communication
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	EP-15		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-43: "Twitter" style notification*		
Description	Receive only one notification with information on the number of #.		
Requirement type	Functional	**Category**	Communication
Priority	Low	**Prototype Version**	Version 2
Fulfilled by	To be specified in V2		

ID: Title	*FR-44: Images in the announcements*		
Description	Announcements can include an image.		
Requirement type	Functional	**Category**	Communication
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	EP-14		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-45: Activities are mapped in a collection of timeslots*		
Description	When you create an activity you set a time range (and this can be edited) and define time slots.		
Requirement type	Functional	**Category**	Activity management
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-22, EP-23		

ID: Title	*FR-46: Timeslots have counters*		
Description	Users should be able to mark their availabilities and their need for each timeslot. Counters can represent volunteer availabilities AND needs (children).		
Requirement type	Functional	**Category**	Activity management
Priority	High	**Prototype Version**	Version 1
Fulfilled by	Frontend, EP-22		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-47: Propose different activities*		
Description	Users should be able to propose multiple activities for the same time slot.		
Requirement type	Functional	**Category**	Activity management
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-22, EP-23		

ID: Title	*FR-48: Problematic timeslots*		
Description	Users should be able to see time slots that are problematic (when the minimum number of volunteers or children is not reached, or when there is an unbalanced ration between volunteers and children, etc.). Using different colors to signal problematic and non-problematic timeslots.		
Requirement type	Functional	**Category**	Activity management
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-22, EP-23		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-49: Export agenda as XLS*		
Description	Export an agenda with all activities in the group calendar for supporting discussion as XLS file (plugin).		
Requirement type	Functional	**Category**	Activity management
Priority	High	**Prototype Version**	Version 1
Fulfilled by	Frontend, EP-49		

ID: Title	*FR-50: Calendar view*		
Description	Users should be able to see calendar with: - PERSONAL VIEW: the activities in which you are involved, and your children is involved in - GROUP VIEW: all the activities of that group		
Requirement type	Functional	**Category**	Activity management Calendar
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	EP-48, EP-49		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-51: Toggle Calendar / List view*		
Description	Users should be able to toggle view as a calendar (month / week) or as a list (a list of timeslots).		
Requirement type	Functional	**Category**	Activity management Calendar
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	BigCalendar		

ID: Title	*FR-52: Optimal solution*		
Description	The application proposes an optimal solution after collecting votes.		
Requirement type	Functional	**Category**	Activity management
Priority	Low	**Prototype Version**	Version 2
Fulfilled by	To be specified in V2		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-53: Set constraints/ variables for slot confirmation*		
Description	When creating an activity set the minimum number of children or volunteers required for each activity.		
Requirement type	Functional	**Category**	Activity management
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP -19		

ID: Title	*FR-54: Who can propose an activity*		
Description	Every group member can propose an activity and a group admin (just one) should agree.		
Requirement type	Functional	**Category**	Activity management
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	EP-47		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-55: Who can modify / confirm and close an activity*		
Description	Only group administrators can modify or close or delete activities.		
Requirement type	Functional	**Category**	Activity management
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-47		

ID: Title	*FR-56: Activities detail*		
Description	When you create an activity, you define the title and the timeslots. Activity description, location, cost and image are optional.		
Requirement type	Functional	**Category**	Activity management
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-19		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-57: Timeslot information*		
Description	Slots display the activity name, you can add a description, they have a counter for participation and a counter for children registered to that activity.		
Requirement type	Functional	**Category**	Activity management
Priority	High	**Prototype Version**	Version 1
Fulfilled by	Frontend, BigCalendar		

ID: Title	*FR-58: Professionals and timeslots*		
Description	If a professional is involved, he/she can be added in the time slot detail (by group administrators).		
Requirement type	Functional	**Category**	Activity management
Priority	High	**Prototype Version**	Version 2
Fulfilled by	To be specified in V2		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-59: Information on a timeslot*		
Description	In a single time slot, you can define the location and the description.		
Requirement type	Functional	**Category**	Activity management
Priority	High	**Prototype Version**	Version 1
Fulfilled by	EP-22, EP-23		

ID: Title	*FR-60: Apply changes to activity time for all the time slots*		
Description	Users should be able to modify time for multiple time slots.		
Requirement type	Functional	**Category**	Activity management
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	Frontend, EP-23		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-61: You can duplicate a time slot (for recurrent events)*		
Description	Repeating a timeslots means repeating for a week/month/same days.		
Requirement type	Functional	**Category**	Activity management
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	Frontend		

ID: Title	*FR-62: Export in your calendar*		
Description	Export an activity calendar (once it is fixed) to your digital calendar.		
Requirement type	Functional	**Category**	Calendar
Priority	Low	**Prototype Version**	Version 2
Fulfilled by	To be specified in V2		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*FR-63: Calendar in the homepage with commitment and fixed slots*		
Description	A calendar showing all the activities in which I am involved (confirmed or open) or my children (confirmed or open) is displayed in the app homepage.		
Requirement type	Functional	**Category**	Calendar
Priority	Low	**Prototype Version**	Version 1
Fulfilled by	Frontend		

ID: Title	*FR-64: Calendar with colors*		
Description	The color represents time slots (days or weekly) colored if there is an activity for me / or my child(ren).		
Requirement type	Functional	**Category**	Calendar
Priority	Medium	**Prototype Version**	Version 1
Fulfilled by	Frontend, BigCalendar		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*NFR-01: Data encryption*		
Description	The SYSTEM shall support encryption of sensitive data at rest and in motion. Sensitive data may refer to information about personal data and/or data generated by end-user behaviour. Managing and securing encryption keys is also important. Encryption keys should be stored separately from the data.		
Requirement type	Non-functional	**Category**	Security
Degree of necessity	High	**Prototype Version**	Version 1
Fulfilled by	Transparent Data Encryption will be supported natively by the database modules within containers.		

ID: Title	*NFR-02: Backup*		
Description	The SYSTEM shall be able to ensure the data integrity with frequent backups. All data will be replicated in remote systems in Athens and Thessaloniki in Greece. Data are backed up daily and a backup copy is stored offside but again in data centers in Athens and Thessaloniki in Greece.		
Requirement type	Non-functional	**Category**	Data Integrity
Degree of necessity	High	**Prototype Version**	Version 1
Fulfilled by	The databases will be subject to daily backups.		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*NFR-03: Access management – User Authentication*		
Description	The SYSTEM shall support authentication measures. Only authenticated platform users shall be able to login to the platform and use its resources except those specifically intended to be public. Authentication will be compliant with the European Parliament Directive 2002/58/EC.		
Requirement type	Non-functional	**Category**	Security
Degree of necessity	High	**Prototype Version**	Version 1
Fulfilled by	Frontend, EP-41		

ID: Title	*NFR-04: Access management – Access Control*		
Description	The SYSTEM shall provide an access control mechanism (e.g role-based access control - RBAC) for restricting system access to authorized users (authorization). That is in order to provide the ability to differentiate users according to their roles in communities, groups and activities, as specified in the related functional requirements. In relation to API management, handling of exposed APIs including control of which interfaces are offered to which users is also foreseen.		
Requirement type	Non-functional	**Category**	Security
Degree of necessity	High	**Prototype Version**	Version 1
Fulfilled by	Frontend, appropriate database schema and user roles		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*NFR-05: Performance - Auditing*		
Description	The SYSTEM shall support the capability of collecting logs for monitoring and auditing purposes.		
Requirement type	Non-functional	**Category**	Auditability
Degree of necessity	High	**Prototype Version**	Version 1
Fulfilled by	Native support by the deployment environment		

ID: Title	*NFR-06: Privacy – Anonymous Reporting*		
Description	The SYSTEM must ensure that it is possible to provide anonymous reporting of analytics data such that the privacy of individual end-users is not compromised.		
Requirement type	Non-functional	**Category**	Auditability Privacy
Degree of necessity	High	**Prototype Version**	Version 1
Fulfilled by	Native support by the deployment environment, database encryption (NFR-01)		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*NFR-07: Personal data removal*		
Description	The SYSTEM shall provide a data structure and a corresponding mechanism to allow users to completely remove their data, while ensuring the consistency of the remaining data.		
Requirement type	Non-functional	**Category**	Privacy Data Integrity
Degree of necessity	High	**Prototype Version**	Version 1
Fulfilled by	EP-43, appropriate database schema		

ID: Title	*NFR-08: Sandboxed deployment*		
Description	The SYSTEM will ensure that application instances, each corresponding to a separate community, will be deployed in a sandboxed manner, ensuring isolation from one another.		
Requirement type	Non-functional	**Category**	Deployment
Degree of necessity	High	**Prototype Version**	Version 1
Fulfilled by	Native support by the deployment environment		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*NFR-09: Multilingual support*		
Description	The SYSTEM will ensure inherent support of multiple languages, with proper categorization and organization of the content, keeping language-specific resources separated from the rest of the application.		
Requirement type	Non-functional	**Category**	Localization
Degree of necessity	High	**Prototype Version**	Version 1
Fulfilled by	Native support in React		

ID: Title	*NFR-10: Scalable deployment architecture*		
Description	The SYSTEM will ensure that new application instances can be deployed easily, with no implication for the ones already running.		
Requirement type	Non-functional	**Category**	Deployment Scalability
Degree of necessity	High	**Prototype Version**	Version 1
Fulfilled by	Native support by the deployment environment		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*NFR-11: Data storage*		
Description	Data will be stored at the central protected storage facility of a cloud service provider in Greece, administered by a responsible person from VILABS and certified data protection officer from the Cloud service provider and following the best practices and standards available. The Cloud service provider protected storage facility will be based on redundant systems and located in Athens and Thessaloniki in Greece. Data storage and protection procedures are compliant with 2016/679 (General Data Protection Regulation, GDPR) and European Parliament Directive 2002/58/EC.		
Requirement type	Non-functional	**Category**	Security
Degree of necessity	High	**Prototype Version**	Version 1
Fulfilled by	An appropriate cloud service provider will be chosen and given directions.		

ID: Title	*NFR-12: Privacy consent*		
Description	The SYSTEM should support a privacy consent mechanism, denying access to users that do not provide it. In particular, children information are provided by parents and data are collected according to the GDPR (article 8). This is related to FR-12, FR-14, FR-15 and FR-16.		
Requirement type	Non-functional	**Category**	Deployment Scalability
Degree of necessity	High	**Prototype Version**	Version 1
Fulfilled by	Frontend, NFR-03, NFR-04		

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

ID: Title	*NFR-13: Access to personal data*		
Description	At any moment, and in accordance with the GDPR articles 14, 15, 16 and 18, the SYSTEM should provide interested parties with access to their personal data, including the rights to rectify and remove it. This is related to FR-09, FR-10 and FR-11, as well as NFR-07.		
Requirement type	Non-functional	**Category**	Privacy
Degree of necessity	High	**Prototype Version**	Version 1
Fulfilled by	NFR-07, EP-40, EP-43		

5 Fulfillment of ethics requirements

Through the Families_Share platform, many people will give their personal information out on the internet and the platform will keep track of them, saving them in an encrypted protected storage, to be displayed into fully anonymized platform analytics and KPIs. In order to ensure compliance with ethics requirements corresponding deliverables with guidelines have been released in Month 3 under WP7. In this section we show how these guidelines are taken into account in the platform development with respect to data management:

- Compliance with D7.1 – Ethic requirement No. 3, is ensured by *NFR-11: Data storage*, *NFR-01: Data encryption* and *NFR-03: Access management – User Authentication.*
- Compliance with D7.3 – Ethic requirement No. 5, is ensured by the requirement *NFR-12: Privacy consent.*
- Compliance with D7.5 – Ethic requirement No. 7, is ensured *NFR-06: Privacy – Anonymous Reporting, NFR-07: Personal data removal, NFR-13: Access to personal data.*
- Compliance with D7.6 – Ethic requirement No. 8, is ensured by *NFR-01: Data encryption, NFR-02: Backup* and *NFR-03: Access management – User Authentication, NFR-04: Access management – Access Control, NFR-11: Data storage, NFR-12: Privacy consent*

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

6 Conclusions and future work

This document outlines the work done under Task 2.1. It is closely linked with D1.2 and provides the basis for the first version of the Families_Share platform, to be described in D2.2. To that end, we presented a consolidated list of the formalized functional and non-functional requirements. This was followed by a specification of the platform architecture, its software components and the deployment environment. Moreover, the API back-end was specified, in the form of RESTful endpoints. Finally, we provided a correspondence between the requirements and the aforementioned specifications to demonstrate how the former will be satisfied by the latter.

The next step is clearly to focus on the implementation of the first prototype, based on the guidelines of this document. We are confident that the implementation will comply with the design requirements, mainly because of two main strengths:

- Building on well-established and state of the art technologies for software development, but with a wide community and a variety of libraries and add-ons to ensure both flexibility and robustness.
- Using a dockerized deployment architecture to strive for security and flexibility among the different CityLabs.

Moreover, close collaboration with WP1 will be needed to ensure best possible adaptation to the user needs, proper localization of the apps and preliminary testing before the pilots start in March 2019.

This project has received funding from the European Union's Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783

7 References

[1] “AngularJS”. [Retrieved October 2018]. [Online]. Available: https://angularjs.org/

[2] “Laravel”. [Retrieved October 2018]. [Online]. Available: https://laravel.com/

[3] “Angular”. [Retrieved October 2018]. [Online]. Available: https://angular.io/

[4] *"Apache Cordova". [Retrieved October 2018]. [Online]. Available:* /https://cordova.apache.org/docs/en/latest/

[5] *"MERN Stack". [Retrieved October 2018]. [Online]. Available:* http://mern.io/.

[6] *"Move over LAMP". [Retrieved October 2018]. [Online]. Available:* https://simantas.wordpress.com/2016/09/09/move-over-lamp-its-either-mean-or-mern/.

[7] *"MongoDB". [Retrieved October 2018]. [Online]. Available:* https://www.mongodb.com/.

[8] *"ReactJS". [Retrieved October 2018]. [Online]. Available:* https://reactjs.org/.

[9] *"Node.JS". [Retrieved October 2018]. [Online]. Available:* https://nodejs.org/en/.

[10] *"Javascript everywhere". [Retrieved October 2018]. [Online]. Available:* https://tinyurl.com/l59uy2r.

[11] *"Express.JS". [Retrieved October 2018]. [Online]. Available: https://expressjs.com/.*

[12] “Docker”. [Retrieved December 2016]. [Online]. Available: https://www.docker.com/

[13] Richardson, L., & Ruby, S. (2008). RESTful web services. “O'Reilly Media, Inc.”.

[14] Christensen, J. H. (2009, October). Using RESTful web-services and cloud computing to create next generation mobile applications. *In Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications* (pp. 627-634). ACM.

[15] Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. *Linux Journal*.

[16] “GitLab API documentation”, [Retrieved December 2016]. [Online]. Available: https://docs.gitlab.com/ee/api/

[17] “MEAN Stack”. [Retrieved October 2018]. [Online]. Available: https://www.mean.io/

[18] “Python”. [Retrieved October 2018]. [Online]. Available: https://www.python.org/

[19] “Cyclopt JS starter kit”. [Retrieved October 2018]. [Online]. Available: https://github.com/cyclopt/js-starter-kit

[20] “FullCalendar”. [Retrieved October 2018]. [Online]. Available: https://fullcalendar.io/

[21] “BigCalendar”. [Retrieved October 2018]. [Online]. Available: https://github.com/intljusticemission/react-big-calendar

[22] “Ant Design”. [Retrieved October 2018]. [Online]. Available: https://ant.design/docs/react/introduce

[23] “Ant Design Calendar”. [Retrieved October 2018]. [Online]. Available: https://ant.design/components/calendar/

This project has received funding from the European Union’s Horizon 2020
CAPS Topic: ICT-11-2017, Type of action: IA, Grant agreement No 780783