Work Web-chat Application

# Pokémon Rescue Squad

- Dávid Lázár, Karthik Devarajan, Matija Rupčić, Simone Tron

# Specifications

## Purpose of the application

We are going to develop an intuitive chat application, which would be scalable and easy to use. The Chat application will be added to a Document Management System - DMS. DMS is already developed and used in Orange.

## Goals and requirements

* Makes communication among employees in a company easier, faster, and more effective.
* Provides user friendly and safe work application where users can review files and chat simultaneously.

1. Makes possible for employees to share documents and files
2. Makes possible for employees to comment regarding specific documents and projects.

* Provides layouts which separates project conversations and documentation from the common ones
* Provides efficiant and clear notification system about the deadlines and important notes.

## Users

End users of this application are people from working environment i.e. **Employees of a company**.

There are two types of users:

1. **Manager** one who has to handle a lot of documentaiton, manage teams, and lead project proccesses.
2. **Other employees** who needs to be up-to-date with information and documentation about projects and job tasks.

*The application will be modified for working environment. Regardless, the user could be anyone who is working with a lot of documentation and on projects. This includes various teams and groups from freelancing environment to educational environment.*

## Competition analysis

There are few similar applications, designed for co-workers communication, and file sharing, where some of them are more oriented to communication, and the other for file sharing and work tasks.

**Communication oriented competitors:**

[***Slack***](https://slack.com/) – the biggest competition regarding chat environment. User friendly layout and features. This app will partly be our UI inspiration. Our client specifically asked for Slack inspired chat environment, especially regarding features. We will rely on similar ideas when it comes to chatroom development.

[***Telegram***](https://telegram.org/)- easy to use chat app, with very user-friendly mobile version. Some chat features could be used as inspiration for chatroom features.

**File sharing and work tasks-oriented competitors:**

[***Microsoft Teams***](https://www.microsoft.com/en-us/education/products/teams/default.aspx) – the biggest competition regarding work files sharing and communication. App has some interesting and useful features, such as files manipulation, team creation, meetings schedule. The logic of work-related concept could be used and modified. We consider the layout of this app not as user friendly as it could be.

[***Flock***](https://flock.com/) – Similarly as MT, the app was created specifically for working environment. It also consists interesting file-sharing and team creation features. This app doesn't give much attention to job tasks and schedules.

[***Ryver***](https://ryver.com/) – Similar to Flock, but with much more focus on tasks and very well-made notification system. This app lacks chat and communication features.

Most of the competitors are very expensive to use with all the features. Free versions are available in some cases, but are limited for everything that work environment needs. All of the competitors are more focused on one type of work environment collaboration, and are lacking equal representation of chatting and project space and a flow between them. In our application, we will try to combine both, and to create a flow between communication and project organization

## Functionalities

These functionalities will be implemented and supported thoroughly in this project:

### Functional requirements

1. **User registration/login** – every user will have to register first, and then login every time when he/she uses the application. By registering , users can create their profile, where they can edit before or after registration. Profiles will include some demographic information, work position information, profile picture and other details about the user.
2. **Peer-to-peer chatting** – users will be able to start conversations with other users and chat.
3. **Groupchatting** – users will be able to create group conversations with other users and chat with each other.
4. **Privacy and Privileges management** – users will have a possibility to customize their privacy settings, but also, administrators of group chats or project discussions will have the right to manage additional privacy settings, such as who is allowed to see a documents or to join discussions. Also, administrators will be able to moderate the conversations and project content more thoroughly than other users.
5. **Projects management and discussion** – the app will enable users to create and edit information about projects, share and discuss files, upload and download files, search the files and conversations, and create discussions for projects only.

### Non-functional requirements

1. ***Intuitiveness*** - the app will be easy to use and learn for the users
2. ***Scalability*** - the app will be scalable, thus any additional changes or features will be easy to implement
3. ***Privacy*** - Administrators will only be able to see necessary data about users (passwords will be encrypted and stored in a non-accessible part of the database)
4. ***Security*** (possibly) – All chat messages will be of end-to-end encryption.

### Technologies

* **HTML5**, **CSS**
* **Vue.js** - simple and yet complex JS framework, which scales well, is fast and competitive compared to Angular and React
* **Vuex** - statefull service for Vue.js, state management pattern and library
* **Material Design** - used in DMS, intuitive, good-looking, a undiscussable requirement
* **Firebase** and **Firestore** - NoSQL database, works well for small projects.

Estimated Expenses

|  |  |
| --- | --- |
| Budget Line Item | Cost Range |
| Site Design & Coding | $1000-$2000 |
| Site Architecture ,Wireframes & Templates | $1000-$2000 |
| Content Planning and Development | $500-$2000 |
| Web Hosting and Web Integration | $1000-$3000 |

## Schedule

### Specification:

**Submission**: 21.10.2018

*Submit project's Specifications*: goals, functionalities, schedule and responsibilities, 2 different personas, use-case for each with before and after scenario, and wireframes and storyboards for every use-case

**Peer-review:** 24.10.2018

**Improved submission:** 28.10.2018

**Team review:** approx. 31.10.2018

### Prototype development and testing

**Submission:** 11.11.2018

**Peer-review**: 14.11.2018

**Improved submission:** 18.11.2018

**Team review:** approx. 21.11.2018

### Layout and functionalities

Approx. early December 2018

### Content

Approx. the end of December 2018

## Responsibilities

### Dávid Lázár

Backend and Frontend development

### Karthik Devarajan

Backend development, Project documentation

### Simone Tron

Frontend development, Project documentation

### Matija Rupčić

UX/UI Design, Contents

## 

## Personas

## 



## Use-cases

### Desktop use-case

#### Before scenario

Adam is swamped with work. Suddenly, he remembers that deadline for one of the projects with his Marketing team , working on is coming soon. He has to gather all the people working on the project and wants to discuss some important points with them.

He searches his computer for project documentation. He calls or sends email to people in the team and sets up a meeting in 2 hours. He prepares short presentation about important points and brings the rest of the documentation with him.

Once everyone’s gathered they start discussing. The time is limited, considering some of the co-workers have another meeting after. Also, one of the coworkers isn’t at work today, so Adam must contact him as soon as possible.

#### After scenario

Adam is swamped with work. Suddenly, he remembers that deadline for one of the projects with his Marketing team, working on is coming soon. He has to gather all the people working on the project and wants to discuss some important points with them.

He logs in the WorkChat and creates a conversation for people who are working on the project, by sending them invitations to join the conversations. He arranges for the Project description and puts the most important points.

He starts a discussion with co-workers, referring to the important points. One of the co-workers is absent, but he will get all the relevant information once he logs into the application.

### Mobile use-case

#### Before scenario

Marek is traveling. His boss contacts him for a review of a project documentation and a signature. Marek is busy, but it is urgent to deal with what his boss has asked.

He asks the boss to send him documents via email. There is a security issue when it comes to a file that has to be sent but email isn't the best option because enterprise documentations are confidential. Also, other colleagues reviewed the document and some of the points they commented on, might be important for Marek’s review.

Marek downloads document on his phone, reads it as fast as he can. He's missing comments of coworkers but manages to spot and to think about crucial points. He signs the document on the phone, and sends it to a boss, with both of them being aware of the security risk. He includes his commentary in the email body.

#### After scenario

Marek is traveling. His boss needs him to review a project documentation and to sign it. Marek is busy, but it is urgent to deal with what his boss has asked.

Marek got a notification regarding the project and he opens the WorkChat app. He sees it's about important project, and he also notices a deadline alert. Inside a conversation that he opened due to notification, there is an assignment for him to check the file attached to it, and sign it once he reviews it. Also, he can find the suggestions and comments made by the other colleagues about the same document.

He opens the document and reads it as fast as he can. He signs it and saves it. He then uploads the new file version to the Project section. He then puts a few comments about the exact file, after double-clicking the file icon. Considering he read the notification and did the marked file, his boss gets a notification that Marek has accomplished what he was asked for.

## Wireframes and Storyboards

* Desktop use-case [Linkto PDF](https://drive.google.com/open?id=1oZM7j30ImSHlcu3Pt4pzA7NRJfeef90k)
* Mobile use-case [Linkto PDF](https://drive.google.com/open?id=1M5V5K3QIR4KdKWA_PHeKEIWMDL8pcUAQ)