**Project overview - Technical Documentation**

Summary :

1. What is Star Citizen
2. What’s the purpose of “StarCrew” ?
3. Work accomplished / to do
4. Development guidelines
   1. Installation process
   2. Tech stack
   3. General functioning

# **1.What is star citizen ?**

Star citizen is a crowd-funded game under development directed by Chris Roberts, maker of the Wing Commander series.

Star Citizen can be described as a Space Simulator where players can fly in starships as a crew or alone and explore on foot planets, asteroids or space station.  
The goal of the game can differ depending on the player.  
One might want to ferry cargo between two location, one other might want to attack this player in order steal this cargo. And some would want to raid a pirate base in order to reclaim the stolen wares.  
 To recap, Star Citizen has a lot playstyles to offer and all of them can require several players to be properly experienced. Thus, player will need to find other players in order to achieve their goals and this is where “StarCrew” appears.

# 

# 

**2.What’s the purpose of “StarCrew” ?** “StarCrew” is an application that allows player from around the world to find a crew for their mission depending on the language they speak, when they play and what they want to achieve.

A user can use “StarCrew” in two different ways, he can create his own mission and recruit other players or apply for other people’s missions and join a pre-existing crew.

As said earlier, a user has access to two main features : creating a mission and applying for a mission. The following will explain the functioning of these features

# **3.Work accomplished / to do**

A summary of all the features planned and done is available in the file “State of the project”.

Those features are non contractual, the feature list is subject to changes (and likely to receive new features)

Some features are not implemented but already modelized, those models can be found in the bill of specifications.

# **4.Development guidelines**

## a.Installation process

**Laravel :**

The following documentation will help you to install Laravel 5.2 <https://laravel.com/docs/5.2>

Once Laravel is properly installed, you can import StarCrew in your web folder (www or an alias)

Link to the github repo : <https://github.com/rafatic/starcrew>

**Configuring the app :**

In the root folder of the application, the .env allows the configuration. Here are the mandatory points to cover :

APP\_KEY : You will need to generate your own key by running the following command :

php artisan key:generate

This command will generate a new key in the .env file

DB\_CONNECTION : This is used to define which DBMS you want to use.

Currently, Laravel supports the following options :

mysql : MySQL

sqlsrv : Microsoft SQL Server

sqlite : SQLite

postgres : Postgres server

The following documentation will help you configure the database access

<https://laravel.com/docs/5.2/database>

The rest of the .env file is not relevant since the application doesn’t use Redis nor Mail server

**Migrating the database :**

Before adding the data itself, you will need to create the database and give it the appropriate name (the one written in the .env file). The encoding needs to be in utf-8.

Finally you will need to run the following command to add the different tables.

php artisan migrate

Basic data is available in the seeder; To seed the database, use the following command

php artisan db:seed

This will add default languages, roles and mission goals to the database.  
If you want to add data in the seeder, the seeding classes are available in /database/seeds

## b.Tech Stack

**Server :**

The current production server uses NGINX, since laravel relies on .htaccess files, it doesn’t work natively with NGINX, some configurations have been made to replicate the functioning of the .htaccess.

**Back end :**

PHP : Laravel 5.2

DB : MySQL (can be easily changed by configuring laravel)

**Front end :**

Javascript : JQuery

plugin :

* [DateTimePicker](http://xdsoft.net/jqplugins/datetimepicker/)

CSS : Bootstrap

Documentation generation : [PHPDocumentor](https://www.phpdoc.org/)

Development norms : [PSR-2](https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-2-coding-style-guide.md) [PSR-4](https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-4-autoloader.md)

## c.General functioning

The application is built following a standard MVC model. Since Laravel provides a powerful ORM, the models are not heavily used in this application. Thus, we mainly interact between controllers and views. For the authentication system (login/register), we use Laravel’s Auth middleware. All the other components are hand made.