1. Concept
   1. User Interface

The following chapters describe a raw layout concept of the application’s pages.

* + 1. Public - Home page

The Home page is displayed when entering the base URL of the application, hence acting as a landing page. As displayed below, it consists of three panels:



Figure 1 - Home page

* **Welcome Panel:** contains the application logo and a short welcome message.
* **Login Panel:** besides a standard *Login form*, containing of username and password fields, it also contains of a *Register form* with fields for email, username, password and password confirmation. Only one form can be display at a time, toggled in an accordion effect.
* **Link Panel:** contains several links, e.g. to a demo area, tutorial area or news.
  + 1. Protected - User Main Menu

The User Main Menu is displayed on all pages that are protected. It consists of the following parts:



Figure 2 - User Main Menu

* **Logo**: the icon or symbol of the Application
* **Home**: leads to the User Home Page
* **Match**: contains a submenu for creating, looking for matches etc.
* **Language**: submenu for selection of the display language
* **Account**: leads to the User Configuration Section
  + 1. Protected - User Home Page

The User Home Page ist the default landing page after successful login. It contains data, which is relevant for the user, such as:

* New messages
* New match invitations
* User statistics

The list is not yet final.

* + 1. Protected - User Configuration Section

The User configurations can be accessed via the User Main Menu. It is divided into three sections:



Figure 3 - User configurations

* **Account Data Panel**: contains inputs for user account relevant data, such as the user’s E-Mail or fields to set a new password.
* **User Data Panel**: contains inputs and tools to configure the user’s global data, such as the user name, his avatar, etc.
* **Game Settings Panel**: contains tools forsetting properties for the matches, e.g. a preferred colour or map theme.
  + 1. Protected - Match Area
  1. Model

1. Setup

The following chapter describes in a general way, how the project setup was realized on a Windows 7, 64bit system.

* 1. Local Runtime Environment

This chapter describes how to setup *Laravel* ***Homestead*** which is used as the runtime environment for development. Explanations:

* **Virtual Box:** Virtualization engine, that allows to setup, run and administrate virtual machines.
* **Vagrant:** Tool for creating and administrating virtual machines. Virtual machines can easily be setup by importing a “Vagrant box”. A Vagrant box is a set of configurations for a virtual machine (e.g. for the OS or installed software on the VM).
* Laravels **Homestead** is a pre-configured Vagrant box. For the details of Homestead (e.g. the preconfigured OS, webserver or database, see the homestead homepage).
  + 1. Prerequirements

The following configurations must be done before starting the setup:

1. Hardware virtualization enabled in BIOS

|  |
| --- |
| *enabled by default* |

The following software must be installed before starting the setup:

1. Git

[Git Bash] After Installation, configure the user name and the user email:

|  |
| --- |
| 1 git config --global user.name "my\_user\_name"  2 git config --global user.email my\_email@mail.net |

1. Virtual Box (required for Vagrant)

|  |
| --- |
| File: VirtualBox-4.3.20-96997-Win.exe  Version: 4.3.20 for Windows hosts x86/amd64  Install Path: C:\Program Files\Oracle\VirtualBox |

1. Vagrant

|  |
| --- |
| File: vagrant\_1.7.2.msi  Version: 1.7.2, for Windows  Install Path: C:\Program Files (x86)\HashiCorp\Vagrant |

1. Microsoft Visual C++ 2012 Redistributable (x64) (required for PHP)

|  |
| --- |
| File: entwicklung/installer/vcredist\_x64.exe |

1. PHP (required for Composer)

Extra configurations:

* + Put PHP’s Install Path on the system path.
  + The shipped file php.ini.development must be renamed to php.ini
  + PHP’s extensions directory is pointing to C:\php by default. If necessary, change this in the php.ini:

extension\_dir = "ext"

* + To enable Composer downloading resources via Open SSL, enablethe according PHP extension in the used *php.ini*:

extension=ext/php\_openssl.dll

(Note that the location of the .dll file must be relative to the php.ini file)

* + Enable the PHP mbstring extension in the used *php.ini*:

extension=ext/php\_mbstring.dll

|  |
| --- |
| File: php-5.6.5-nts-Win32-VC11-x64.zip  Version: 5.6.5, for Windows 64bit  Install Path: D:\privat\Entwicklung\tools\php\_5.6.5 |

1. Composer

Extra configurations: Put Composer’s *vendor/bin* directory on the system path.

|  |
| --- |
| File: Composer-Setup.exe  Configurations: Composer’s *vendor/bin* diretory: C:\Users\fvo\AppData\Roaming\Composer\vendor\bin |

* + 1. Installation

The following steps describe how to install *Homestead* and *Laravel*.

1. [Commandline] Add the box “laravel/homestead” to Vagrant:

|  |
| --- |
| vagrant box add laravel/homestead |

1. [Git Bash] Download Homestead manually with Git to the current directory:

|  |
| --- |
| git clone https://github.com/laravel/homestead.git |

1. [Commandline; path of the downloaded Homestad instance] Initialize the cloned Homestead:

|  |
| --- |
| init.sh |

This creates the directory *[user home]\.homestead*, including the *Homestead.yaml* configuration file.

1. [Git Bash] Install the *Laravel Installer* via Composer:

|  |
| --- |
| composer global require "laravel/installer=~1.1" |

1. [Commandline] With the Laravel Installer, a new Laravel project can be installed in the current directory (“my-project” is from now on used as example name of the Laravel project):

|  |
| --- |
| laravel new my-project |

* + 1. Configuration

A proposed configuration for Homestead in the *Homestead.yaml* file:

|  |
| --- |
| ---  ip: "192.168.10.10"  memory: 2048  cpus: 1  authorize: ~/.ssh/id\_rsa.pub  keys:  - ~/.ssh/id\_rsa  folders:  - map: [my-project home on local drive]\src  to: /home/vagrant/my-project/src  sites:  - map: virtualhost.app  to: /home/vagrant/my-project/src/public  databases:  - homestead  variables:  - key: APP\_ENV  value: local |

To map the above configured domain “virtualhost.app” to the above configured IP address 192.168.10.10, add the following entry to the system’s *hosts* file:

|  |
| --- |
| 192.168.10.10 virtualhost.app |

* 1. Version Control

For local repositories, Git is used on the local system. For remote repositories, *GitHub* is used. For this, a present GitHub account is required.

1. [Git Bash] Create SSH-Keys for the user configured in local Git (see 1 in 1.1.1):

|  |
| --- |
| ssh-keygen -t rsa –C „my\_email@mail.net“ |

This creates a private and a public part of the SSH key. You will also be prompted for a passhrase. Chose one and note the chosen passphrase. Subsequently the files [user home]/.ssh/id\_rsa.pub and [user home]/.ssh\id\_rsa are changed/created.

1. [Git Bash] Save the public key part on GitHub in the account’s settings. Add the private key for the SSH client:

|  |
| --- |
| exec ssh-agent bash  ssh-add .ssh/id\_rsa |

The generated SSH public key part also hast to be saved on GitHub. Therefore, copy the generated

1. [Git Bash] Check the setup with exactly these parameters:

|  |
| --- |
| ssh –T git@github.com |

* 1. Development Environment
     1. SQL Tool

In the following, modification of the SQL client *Squirrel* is described. It is assumed, that Squirrel is already installed on the local machine:

1. Install the MySQL connector.

|  |
| --- |
| File: mysql-connector-java-5.1.34-bin.jar |

1. Configure and save a connection to the Homestead database. The default SQL connection credentials are:

Connection String: jdbc:mysql://127.0.0.1:33060/homestead

User: homestead

Password: secret

1. Optionally, additional databases can be created by logging in to Vagrant with SSH:

|  |
| --- |
| mysql -u homestead –p |

and executing the SQL script

|  |
| --- |
| CREATE DATABASE my\_new\_database; |

It will be accessible with the same credentials as the default homestead database.

* + 1. IDE

In the following, the installation and configuration of IDE *Netbeans* is described.

|  |
| --- |
| File: netbeans-8.0.2-php-windows.exe  Version: 8.0.2 |

1. **Project integration:**

Go to “File” -> “New Project”.

* Choose Project: Select “PHP” -> “PHP Application from Existing Sources”.
* Name and Location: Browse to the Laravel <project home> directory for the entry “Source Folder” and select a “Project Name”.
* Run Configuration: no configuration required.

Additionally, go to “Tools” -> “Plugins”. Select the tab “Settings”, click the “Add” button and add a new Update Provider with the Update Center URL <https://java.net/downloads/nb-laravel-plugin/updates.xml>. Now you can install Laravel plugins from the tab “Available Plugins”

1. **PHPUnit configuration:**

Right click on project and select „Properties“ -> „Testing“:

* Check „PHPUnit“ as test provider
* Add <project\_home>/src/tests for the Test Directories
* „Testing“ -> „PHPUnit“: Check „Use XML Configuration“ and select <project\_home>/src/phpunit.xml as XML configuration file

1. **Vagrant Plugin configuration:**

Install the Vagrant plugin from „Tools“ -> „Plugins“.

Open „Tools“ -> „Vagrant“ -> „Vagrant Options“. Under „General“ enter the Vagrant executable as „Vagrant Script“ (e.g. C:\Program Files (x86)\HashiCorp\Vagrant\bin\vagrant.exe).

Right click on project -> „Properties“ -> „Vagrant“. Enter the directory of homestead’s Vagrantfile as „Vagrant root“ (D:\privat\Entwicklung\tools\homestead).

Now you have full access on the Vagrant commands like “vagrant up” etc. or SSH into the configured Homestead box (default values are host:*127.0.0.1:2222*, user:*vagrant*, password: *vagrant*).

1. **Debugging configuration:**

Due to technical limitations with Netbeans and XDebug on the present platform, no debugging is configured.

1. Development Log

Artisan migrations (maybe also all requests involving DB connections): the DB host property (usually “DB\_HOST”) in the *.env* file may not contain the port.

Test Users (Experimental):

|  |  |  |
| --- | --- | --- |
| Name | E-Mail | Password |
| Testuser | [test@test.de](mailto:test@test.de) | password |
| Testuser2 | [test2@test2.de](mailto:test2@test2.de) | password |