

# Requirements Document

## 1. Introduction

The name of the project is “Automated system of orders” or ASOO. It helps you order with the web a sheet or reference and take away, which already have printed.

The index page has header, footer and content. Header should have buttons for users login and registration, logo. Footer are: navigation bar and logo. Content can have total amount of the orders and scheme "How it works?".

The website will have for login and registration, F.A.Q. page, account page for a faculty and a student, managing system to manipulate orders data and other features.

The website isn't intended for communication between students and manager of faculty.

## 2 User Requirements

### 2.1 Software Interfaces

Frameworks:

- 1) Laravel Framework (PHP);
- 2) VueJS (Javascript);
- 3) Bootstrap 3 (CSS).

Database modules:

- 1) MySql;
- 2) Eloquent ORM.

### 2.2 User Interfaces

User interfaces are presented as attachments to project like mockups.

### 2.3 User Characteristics

Groups of the intended users of the product are:

- 1) students;
- 2) secretaries of dean's office.

### 2.4 Assumptions and Dependencies

- 1) user operating system;
- 2) look of design.

## 3 System Requirements

### 3.1 Functional Requirements

The project will have such options as:

- 1) Registration will be available only for:
  - a. Faculty;
  - b. Student.
- 2) Authorization by:
  - a. Email;
  - b. Not social networks or other methods.
- 3) Reset password by sending email notification.
- 4) Faculty and students have:
  - a. Own account page;
  - b. Contacts;
  - c. Settings;
  - d. Managing orders.
- 5) Faculty functionality are:
  - a. Set up info about faculty;
  - b. Get requests from students to print orders;
  - c. Get email notification after creating a new order for the faculty.
  - d. Changing status of order;
  - e. Managing types for selection of:
    - i. Sheet;
    - ii. Reference.
  - f. Possibility of managing in access to the orders types.
  - g. Download generated file to print.
- 6) Student can:
  - a. Create order for:
    - i. Sheet;
    - ii. Reference.
  - b. View the status of orders;
  - c. Get email notification after changing order status by faculty administrator;
  - d. Fill info about yourself;
  - e. View orders history;
  - f. Create new request with the template from history.
- 7) Student can't:
  - a. Chatting with the faculty;
  - b. Send request before filling own contacts;
  - c. Send request for not available order types.
- 8) Automatic generation of orders by getting info from student account.
- 9) Existing F.A.Q. page.
- 10) Presentable index page with:

- a. Count of all orders
- b. Who we are?
- c. How it works?

## 3.2 Non-Functional Requirements

### 3.2.1 SOFTWARE QUALITY ATTRIBUTES

1. Graphical design should be responsive and adaptive for mobile devices, tablets and desktops.
2. Request from web-server should be quick and less than 0.6ms.
3. Usability and loading page speed should have not less than 80/100 by the Google PageSpeed product.