**Description of Practical Assignment “Personal expenses tracker”**

# Team of Developers

**Commented [DS1]:** Who is developing the system: names and roles in the project.

There are two student participating in the development of the system:

- Patriks Misāns, pm17051

- Georgs Tumans, gt17008

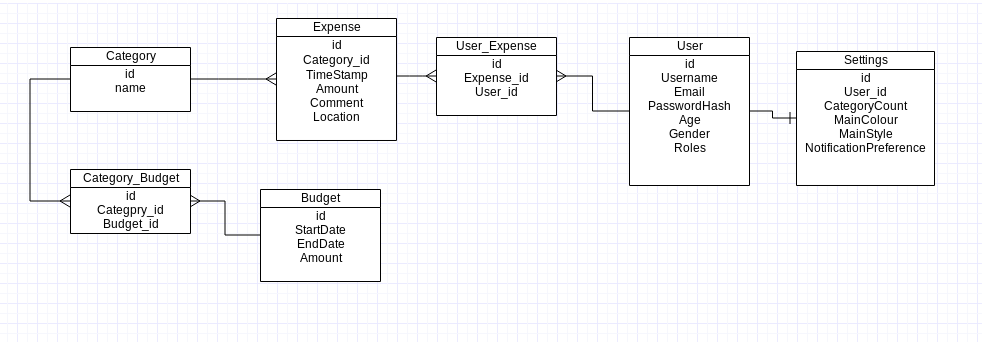
# Development Environment

It is planned to develop the system in *PHP* 7.2.2 environment using the *Laravel* 5.6.11 library. *MySQL* database will be used for data storage with the *phpMyAdmin* system. The code will be stored in the GitHub system.

# Main Functionality

The idea of this project is to develop a web application which is an expenses tracker system. Within this project a website/web application will be developed for entry of data and visualization of the already gathered data. In the future though, there is an idea to also create a mobile app for the system thus massively improving usability of the system in the real life. Going into more details, there are plans to implement categorization of expenses, first, having predefined categories and later creating a system where users can create and manage categories as they wish. An important aspect of the system is display of the expenses themselves, which will be implemented for each of the categories. Among our plans is to show the total expenses as well as expenses within subfields of the big categories. Naturally, the user would have to define the category and the appropriate subfield upon entry of data. We also plan to implement some visual representation of data such as graphs and charts. Moreover, the user will be able to select the time period for expenses statistics. Among the input data, the user could also select the geographical place of expenses, which might also be gathered automatically. In case there is enough time, we also aspire to implement a budget system, which would be helpful tool for living according to a set budget within a set amount of time – the user could enter the sum of money available for expenses within, for instance, a month, and the system would send notifications and reminders in case the available monetary recourses are running out. Some user setting (for the UI perhaps) might also be implemented later on. Finally, among other viable features is a calendar view embedded with expenses.

# Data Registry

The following is the approximate model of the database tables. It consists of the *user*, containing the username, email for notifications, password, role – administrator or user, etc; the *expenses*, *category* and the *budget* tables. The expenses table contains data about the sum of the expense itself and also its time and location so that this data can be used for visualization and overall statistics that can be displayed back to the user. The location system might be more sophisticated, prompting the user to enter the place of living in the beginning only, and then reading location data through the browser and comparing it to the initially entered information – in case it differs largely, user would be prompted to add this new location to the expense.

# MVC

The system will be implemented following an MVC paradigm. The system will be distributed into the following components:

**Models**

* User,
* Expenses,
* Category,
* Budget,
* Settings;

**Views**

* Main view that contains the defined categories, a sidebar and the option to dynamically edit the category view (CRUD),
* User profile view for managing personal settings,
* Category view with visual representations and the CRUD operations,
* An overall statistics view including expenses from all the categories,
* Administration views for viewing users, categories, expenses, budgets and deleting them, overall statistics, etc,
* Budget setup view,
* Some pop-ups,
* Registration/login view.

**Controllers**

* + Main View controller with methods for dynamic definition of categories, perhaps also providing notifications regarding the budget settings,
  + User Controller for CRUD’ing user data,
  + Budget controller – CRUD’ing the appropriate data,
  + Expenses controller – CRUD’ing expenses,
  + Admin Controller with methods for retrieving and showing a list of users, blocking/unblocking/deleting users,
  + Laravel standard Register Controller and Login Controller (perhaps the added functionality of FB login)

# User Roles

The system supports two user roles: registered users and administrators.

Registered user:

* see photos marked as public or where the author is the registered user himself/herself;
* look for photos marked as public or where the author is the registered user himself/herself;
* add comments to the found photos;
* create new galleries;
* add photos to the gallery, edit photos, delete his/her own photos;
* add key words to his/her own photos.

Administrator:

* Change setting of the default look of the page – category list, UI;
* View the overall statistics on users – their location, expenses etc;
* Block/delete users;
* Modify user data upon request.

# User Authentication

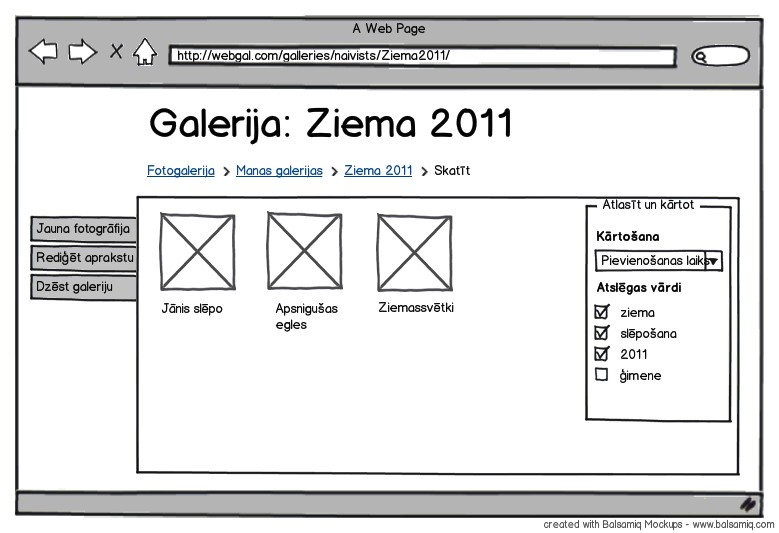
For the user authentication, it is possible to use a local registration system or register using *Facebook* authentication. For users, who have registered through the local registration system, it is possible to link their user to their *Facebook* account.

# System Interface

**Commented [DS6]:** In order to implement this section, it is necessary to better understand the emerging system to visualize what exactly will be generated. This subsequently facilitates programming.

The image was created with “Balsamiq Mockup” free online version: https://balsamiq.com/products/mockups/

The image shows the form for gallery displaying. In the form a user can choose the selection criteria and specify how to arrange the results. Similarly, there are operations available to the user that can be performed with the gallery as a whole.



- 3 -