

CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF INFORMATION TECHNOLOGY

DEPARTMENT OF SOFTWARE ENGINEERING



## POSITIONAL REPORT

### Back-end API and payments system for crowdfunding platform ElateMe

*Kuzmovych Yevhen*

supervised by  
Ing. Jiří CHLUDIL

December 11, 2016

# Contents

1	Introduction . . . . .	2
2	Existing solutions . . . . .	2
3	Possibilities of the solution . . . . .	2
	3.1 Programming language . . . . .	3
	3.2 DBMS . . . . .	3
4	Chosen methods . . . . .	4
	4.1 Python . . . . .	4
	4.2 Django . . . . .	4
	4.3 Django REST framework . . . . .	4
	4.4 PostgreSQL . . . . .	5
5	Current project state . . . . .	5
6	Conclusion and future outlook . . . . .	5

## Keyword

ElateMe, crowdfunding platform, social network, back-end API, DBMS, RESTful

## List of Acronyms

**API** application programming interface

**MVC** model-view-controller

**DBMS** database management system

**REST** Representational state transfer

## 1 Introduction

ElateMe is a new crowdfunding platform with elements of the social network. Unlike other similar projects like Kickstarter or Patreon that help bring creative, commercial projects to life by means of interested people, ElateMe is focused on fulfillment of personal wishes with the help of users' friends. The user can create a wish and set its cost, title, and a short description. His friends then will be able to contribute to his wish by donating money. When wish gathers needed amount, money will be transferred to the user bank account. The social part of the application is providing an ability for the user to subscribe to his friends, communicate with other users, rate and comment others' wishes.

The aim of this thesis is to analyze functional, nonfunctional requirements and use cases of the project, design database model and server architecture, implement back-end application programming interface (API) and payments system for this service.

## 2 Existing solutions

There are many examples of crowdfunding platforms and social networks. But ElateMe is a new concept that combines them both. So there is no exact solution for such system. Yet in this report, popular crowdfunding platforms and social networks will be considered as existing solution. Their stack of technologies can be taken as an example and adjusted for needs of this project.

## 3 Possibilities of the solution

The back-end is part of the application that contains its business logic, processes its data in the database and provides API for the client side of

application (front-end). The main task of back-end developer on initial stage is to select back-end technology stack, therefore, programming language, corresponding web framework and database management system (DBMS).

### 3.1 Programming language

Choice of programming language leads to the choice of the corresponding web framework. It is crucial to choose technology that allows a developer to produce a working solution, supports scalability and flexibility for future modifications and extensions and allows to write elegant, readable and debuggable code.

The most common programming languages and frameworks for web development are (examples of usage were taken from StackShare[1]):

- Ruby and Rails (Kickstarter, Twitter)
- Python and Flask (Patreon) or Django (Instagram)

Ruby is open source, interpreted, object-oriented programming language. It is widely used in web programming. Rails serves as a web framework for the Ruby language. It's used by companies ranging from small start-ups to large enterprises. Ruby on Rails is a great framework for quick prototyping but it can be too slow for large-scale applications.

Python is an interpreted, object-oriented, high-level programming language. Its syntax is simple that makes code written in Python easy to read and understand. It has two main web frameworks that are used by such companies as Patreon, Instagram, Google, Snapchat, and others.

### 3.2 DBMS

Another important part of the technology stack is the database and its DBMS. Considering the fact that structure of this application is highly dependent on data, its integrity, scalability, and security, choice of appropriate DBMS is essential. The most used databases in web development are:

- MySQL (Kickstarter, Patreon, Twitter, Facebook)
- PostgreSQL (Instagram)
- SQLite

SQLite is self-contained, file-based database. It is simple to install and set up. SQLite is often used for debugging purposes as a temporary database.

MySQL and PostgreSQL are the most popular DBMSs for web development. They both offer a lot of functionality to the users. As the standalone database servers, MySQL and PostgreSQL provide powerful interfaces for applications to communicate through.

## 4 Chosen methods

Taken into consideration everything mentioned above and authors own experience and preferences it was decided to use Python as a programming language, Django as a web framework and PostgreSQL as DBMS.

### 4.1 Python

Python will be a base of the server. It was chosen as a primary programming language because it was designed to be simple and highly readable which is very important for large-scale projects. Its syntax and standard library simplify and speed up a development.

The first version of basic back-end API for ElateMe is expected to be ready at the end of January 2017. So fast prototyping is a required option for chosen programming language. That is why Python is the best fit for purposes of this project.

### 4.2 Django

Django[2] is open source web framework for python. It provides a high level abstraction of common web development patterns. It follows model-view-controller (MVC) design pattern. Django uses MVC to separate model a data and a business logic of application, view a representation of the information for the user, in this case, the client side of application and controller a an interface of the application, in this case, set of URLs to communicate with front-end.

### 4.3 Django REST framework

Representational state transfer (REST)[3] is the architectural solution for the transfer of structural data between server and client. It is a broad topic and will not be described in this report. Authors knowledge of REST

comes from *Design solutions for SOAP/WSDL and RESTful web services* by Robert Daigneau[3].

Django REST framework is open source tool built on Django framework. It contains needed tools for implementation of RESTful API and follows all constraints of RESTful server mentioned in [3].

## 4.4 PostgreSQL

On initial stage of the development, SQLite will be used as a DBMS, because it does not require a standalone database server and is simple to set up. The database will be changed and migrated to PostgreSQL later.

PostgreSQL[4] is powerful, open source relational DBMS. It has advanced features such as full atomicity, consistency, isolation, durability. Django framework provides great API for working with PostgreSQL databases.

## 5 Current project state

At the moment the project is at the stage of development. Main business processes, requirements, use cases were analyzed and documented. On back-end side, basic API for user authentication with facebook account and for wishes management has been implemented. On front-end side, there is complete client architecture and partial implementation of the application interface. Next step of development will be studying of online payment systems, a continuation of work on server API, and testing.

## 6 Conclusion and future outlook

The output of this bachelor thesis will be the analysis of business processes, requirements and use cases of ElateMe project, implemented back-end interface for the client-server communication, the online payment systems for donations. The main goal for the author though is to analyze and learn tools for web back-end development such as Python programming language and Django web framework, practice building complicated systems using them, learn to design server architecture and explore various online payment systems.

It is expected to have the working version of the application and its release at the end of academic year 2016/2017. Android and iOS versions of the mobile application will be uploaded to Google Play and Apple Store respectively. There is a possible continuation of work and support of the project in case of successful release and advertising campaign.

# References

- [1] StackShare. *StackShare* [online]. [Accessed 9 December 2016]. Available from: <https://stackshare.io/>
- [2] HOLOVATY, Adrian and Jacob. KAPLAN-MOSS. *The definitive guide to Django: Web development done right*. 2nd ed. Berkeley: Apress, c2009. Expert's voice in Web development. ISBN 978-1-4302-1936-1.
- [3] DAIGNEAU, Robert. *Service design patterns: fundamental design solutions for SOAP/WSDL and RESTful web services*. Upper Saddle River: Addison-Wesley, c2012. Addison-Wesley signature series. ISBN 978-0-321-54420-9.
- [4] OBE, Regina O. and Leo S. HSU. *PostgreSQL: up and running*. Sebastopol: O'Reilly, c2012. ISBN 978-1-449-32633-3.