Tournament Generator

Alexander Büchel

Enrolment number

Fabio Hilti

Enrolment number

Lucy Gannon

Enrolment number

Documentation

University of Liechtenstein

Programme: Masters Programme in Information Systems

Module: Information System Development

Course: Information System Development

Assessor: [Dr. Frank Breitinger]

Working period: 18/09/2020 to 17/12/2020

Date of submission: 17/12/2020

Table of contents

[1 Introduction 1](#_Toc58856111)

[1.1 Project Group 1](#_Toc58856112)

[1.2 Project Description 2](#_Toc58856113)

[2 Project Goals 4](#_Toc58856114)

[3 Requirements 4](#_Toc58856115)

[3.1 Stakeholder requirements 5](#_Toc58856116)

[3.1.1 Requirements Dr. Frank Breitinger 5](#_Toc58856117)

[3.1.2 Requirements end-users 5](#_Toc58856118)

[3.1.3 Requirements project group 5](#_Toc58856119)

[3.2 Functional and non-functional requirements 6](#_Toc58856120)

[3.2.1 Functional requirements 6](#_Toc58856121)

[3.2.2 Non-functional requirements 7](#_Toc58856122)

Abstract

Each type of paper defined in Section 4.1 of the Guidelines for Writing Academic Papers must include an abstract at the beginning. With regard to the structure of an abstract, see Swales and Feak (2009).

Keywords: Keyword 1; Keyword 2; Keyword 3; Keyword 4

# Introduction

On the 9th of October 2020 the first seminar of the course Information System Development (ISD) started, which was also the starting point of the web-based group project. The aim of the project is to develop a web-based project using Phyton/Django. Therefore, skills such as programming, design, creativity, and documentation are required. Before starting with the project, each group member had to get a basic understanding of Phyton/Django with the help of various tutorials, online-videos, and the course’s assessor itself. This document will be part of the documentation, which includes following methodology: project group, project description, project goals, requirements, domain model, user experience and run through, highlights, readme and notes, conclusion.

More Text for Introduction?

## Project Group

Our project group consists of three people: Alexander Büchel, Fabio Hilti and Lucy Gannon. Each of us have little to no experience in Phyton, Django, GitHub, or other web-based applications. For this reason, we had to gain our phyton and coding knowledge almost from scratch. Despite our limited experiences, we had high goals: we wanted to create a functional, useful web-based project. At first, there were various ideas from a language translator right up to tic tac toe. Finally, we have decided to create a Tournament Generator because of its unique idea.

Although each group member tried to work on every project task such as coding, documentation, design, or functionality, we also tried to split up the workload reasonably. Furthermore, our internal communication was probably more difficult than the years before, due to the global corona-pandemic. For this reason, it was not allowed to meet up in person for the project task. The communication occurred mainly online, via Zoom-Meeting or WhatsApp. That is why, we had minor problems with our communication, planning and organisation.

In the following table below, there is shown, which group member had which experiences beforehand and main responsibilities for the web-based group project.

*Table 1: Project Group*

|  |  |  |
| --- | --- | --- |
| **Group member** | **Experiences** | **Responsibilities** |
| Alexander Büchel | * No experience in Phyton/Django * Little experience in coding | * ??? * … * … |
| Fabio Hilti | * Primarily business background * No experience in Phyton/Django | * Design * Documentation * Coding |
| Lucy Gannon | * Little experience in Phyton/Django * … | * ???? * … * … |

## Project Description

Everything can be a competition. To compete with one and each other is in the human nature – it is fun and challenging. All of us three in our group like to participate in different competitions, respectively tournaments in a sport club or just for fun with friends. With that thought in our mind, we developed the idea to create a “Tournament Generator” as our web-based group project in the module “Information System Development” with Python/Django.

With this project, a group of friends, respectively a sport club can simply create a tournament tree/tournament bracket without any complicated prior knowledge. There are already some bracket generators to use on the world wide web, but most of them have unnecessary customization, which will take up a lot of time for the end-user and can be quite tedious. That is why, we as a project group, decided to create our own simple “Tournament Generator” without beating around the bush. The key elements of this web-based project are the three different tournament brackets, which the end-user can choose from: 8, 16 or 32 teams. This means, depending on the tournament size, the tournament bracket will alter themselves on the required size.

For a simple demonstration and understanding on the basic procedure of our web-based project “Tournament Generator”, we created the following flow chart on the following page.

More Text for Project Description?

*Figure 1: Flow chart Tournament Generator*



Any Changes necessary? More or less information in flow chart?

# Project Goals

The goal of this web-based group project is to create a basic Tournament Generator, which – as its name already says – generates different brackets for tournaments with a size of 8, 16 or 32 Teams. In Addition to the brackets/tournament trees, the end-user of should have the possibility to create different teams for their tournaments with several information fields such as team name, number of players, manager, and captain. This teams can be added anytime to the desired tournament. As previously stated, the aim is to create a Tournament Generator for people, who like to be competitive. Hence, the main target group we would like to address with our web-based project are mainly people, who want to organize a spontaneous sporting event without big of a planning for sports such as football, tennis or handball. Moreover, our web-based project should appeal not only for sport-user but rather for competitive fun-tournaments in general (e.g. gaming, Beer Pong, other competitive party-games).

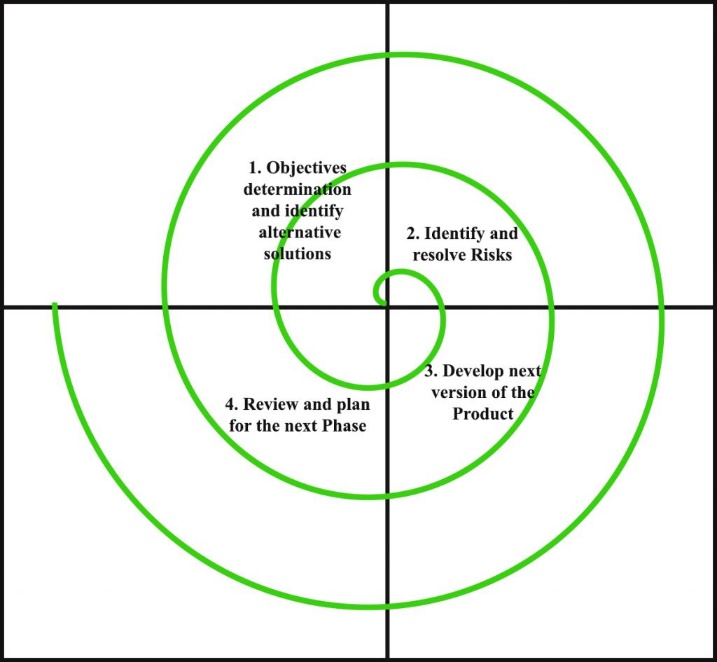
With our own “Tournament Generator” our goal is to offer a basic but interactive website, which will create brackets for the three common tournament sizes that can be used and implemented easily in real life. More Text/Information for chapter Project Goals?

# Requirements

With the project goal in mind, there need to be specific requirements defined. For defining the requirements and process of the “Tournament Generator”, the spiral model for Software Development is used (see Figure 2). The goal of creating requirements is mainly to minimise the risks and gain satisfaction of stakeholders.

The spiral model provides support of risk handling, whereby each loop is called a phase of the software development process. The exact number of phases depend on the group and the project work itself. The 4 phases are “objectives determination and identify alternative solutions”, “identify and resolve risks”, “develop next version of the product” and “review and plan for the next phase” (Kumar Pal, 2018). More Text/Information for chapter Requirements?

*Figure 2: spiral model*



## Stakeholder requirements

As a first step, it is needed to describe the potential and different stakeholders, which also usually have different requirements for the project. For our own web-based projects, the following 3 stakeholders were identified:

1. Dr. Frank Breitinger, Assessor of course Information System Development (ISD)
2. End-users of Tournament Generator
3. Project Group
4. OTHER STAKEHOLDERS???

### Requirements Dr. Frank Breitinger

* A certain minimum of functionality and complexity of the project/website
* Clear documentation of the project including docstrings, readme and examples of usage
* Nice Layout and Design for the project/website
* Code quality of project is well structured and organized
* Chosen project with its features and ideas behind is creative
* Utilization of GitHub or any other control system
* A list of used packages including version and used programming language
* Every member in project group contributes to coding
* More requirements…???

### Requirements end-users

* Web-based project is optimized for all common Internet-Browser such as Microsoft Edge, Google Chrome, Safari and Mozilla Firefox
* Web-based project is simplistic and easy to use and understand, respectively user-friendly
* Web-based project is able to create the three common, different tournament sizes (8, 16 or 32 teams)
* Web-based project allows to create teams, which can be added to various tournaments, including the information fields: team name, number of players, manager, captain)
* Web-based project includes information about the creators of the website and an imprint
* Web-based project guides the end-user on how to easily create a tournament
* Process of the web-based project is simple, quick, and efficient
* Web-based project has a uniform design and is appealing for the end-users
* Web-based project offers a random-autofill, which fills up needed fields with information automatically for an even faster tournament creation
* End-user can put in results of the different matches for deciding the winders of each tournament round
* … more Requirements???

### Requirements project group

* Project group achieves greater understanding in coding, especially in Phyton/Django
* Project group has good internal communication and achieves a great working environment
* Each member of the project group takes responsibility
* Each member of the project group participates in required project tasks such as coding, documentation, or design
* Project group splits the workload of the web-based project fair and evenly
* Different tasks have a common theme (e.g. uniform design, structured GitHub)
* …More requirements project group???

## Functional and non-functional requirements

In addition to the previous stakeholder requirements (see 3.1.), the requirements for our web-based “Tournament Generator” can be categorized in functional and non-functional requirements.

Functional requirements are statements of services the project should provide, how the system should react to specific inputs and how it should behave in particular situations. Furthermore, it can also state, what the system specifically should not do. Non-functional requirements are constraints of the functions such as standards, timing, programming language or development process. They often apply to the system as a whole rather than individual features (Sommerville, 2011). The following functional and non-functional requirements are identified in chapters 3.21 and 3.2.2.

More Text??

### Functional requirements

* Tournament Generator lets end-user create its own individual tournament
  + Create new tournament or choose existing tournament
  + Minimum and Limitation of teams per tournament (until reached full capacity)
  + All created tournaments are displayed on web-based project/website
* Tournament Generator should include a logical path/process from the Creation to the Execution of the created tournament
  + Step-by-step explanation at Starting-Page
  + Process runs like a common thread through tournament creation
* Tournament Generator must include the creation of 3 various tournament brackets
  + Tournament bracket for 8 teams
  + Tournament bracket for 16 teams
  + Tournament bracket for 32 teams
* Tournament Generator should display the winner at the end of each tournament
* Tournament Generator allows end-user to put in match-results to choose the winner of each round
* Tournament Generator enables the creation of teams for tournaments
  + Add or delete teams in tournaments
  + Add information to created teams such as team name, number of players, manager and captain
  + All created teams are displayed on web-based project/website
* Tournament Generator allows random Autofill-function for an even faster creation of tournaments and teams
  + Autofill information for team creation
  + Autofill information for tournament creation
* Tournament Generator contains an About-page
  + Short description of project group
  + Contact data of three group members such as e-mail, adress or telephone
* Tournament Generator contains an Imprint-page
* Tournament Generator has a uniform Corporate design
  + Contains a Footer
  + Contains Navbar
  + Contains Logo
  + Contains Buttons
* Tournament Generator must be precisely documented
  + Documentation
  + Docstring and comments in Phyton/Django project
  + Readme on GitHub
  + Examples of Usage
* Tournament Generator is shared on GitHub
  + Shared GitHub branch contribution
* Tournament Generator adapts to different common Internet Browsers
  + Microsoft Edge
  + Google Chrome
  + Safari
  + Mozilla Firefox
* …other functional requirements???

### Non-functional requirements

* Tournament Generator has appealing and trendy design in the eyes of target end-user
  + Modern and uniform design
  + Neatly arranged and organized
  + Structured web-based project/website
* Tournament Generator must have a high-quality standard and good usability
  + Few to no errors/bugs
  + Adaptability
* Tournament Generator must be easy to maintain and easily usable for end-user
* Tournament Generator has a good data security
  + Inputs from end-users are safe
* Tournament Generator code is well-structured and logical
* Tournament Generator documentation is clear, complete and accurate
  + Documentation on paper
  + Documentation on Phyton/Django via commentary/docstrings
  + Shared GitHub
* Tournament Generator has a good performance and scalability
  + Web-based project can handle higher data input
  + Web-based project returns results fast
* Tournament Generator must have a coherent overall image/impression
* … more non-functional requirements???

## Requirements validation

According to Sommerville there are four different keywords for the checking of the requirements: validity, consistency, completeness, realism, and verifiability (2011). With these validation-keywords our “Tournament Generator” will be reviewed with the question in mind if we have fulfilled the mentioned requirements.

*Table 1: Requirements validation*

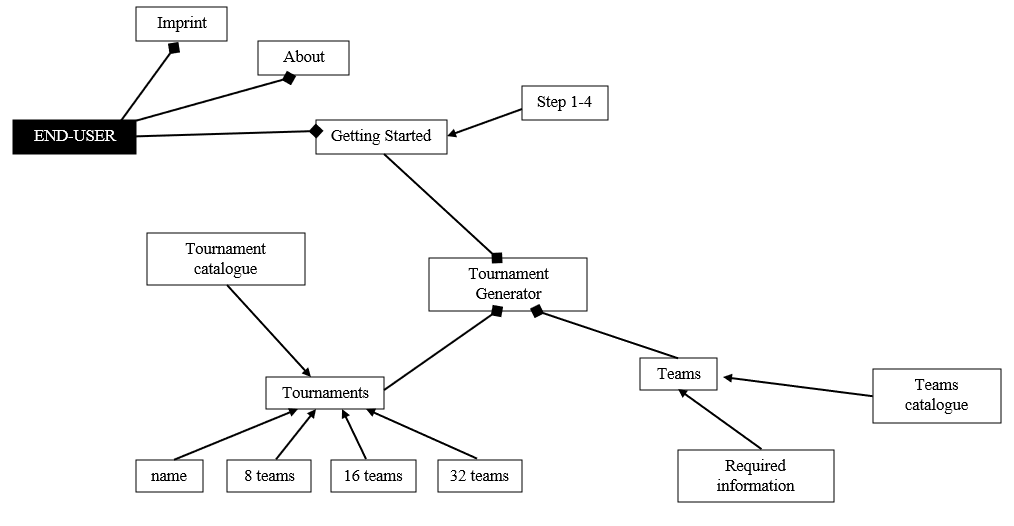
|  |  |  |
| --- | --- | --- |
| **Keyword** | **Clarification** *(Sommerville, 2011)* | **Validation** |
| Validity | Does our system provide the functions which support our user’s needs? | Text Text??? |
| Consistency | Are there any requirements, that conflict themselves? | Text Text??? |
| Completeness | Are all requirements for the project included? | Text Text??? |
| Realism | Can the requirements be implemented with the available budget and technology? | Yes, the requirements can be implemented. For a simple tournament bracket, there is little to no budget as well as special technology besides the standard needed. |
| Verifiability | Is it possible to check the listed requirements? | Text Text??? |

More text for chapter requirements???

# Domain Model

For an optimal and simple overview of the end-user’s most important functional requirements of the web-based project, we created a domain model, which is depicted below on the next page (Figure 3, page 9). For the creation of the domain model we orientated oneself by the example of Rosenberg and Stephen (2007).

*Figure 3: Domain model for Tournament Generator*



# User Experience and Run-through

Text Text Text

## Highlights

Text Text Text

# Readme and Notes

An explicit documentation/commentary of the used code in our web-based project “Tournament Generator” is used with docstrings in Python/Django itself. In Addition to the commentary, there is a Readme available on our shared GitHub.

# Conclusion

After finishing the project, we want to retrospect and evaluate what went well and what went rather wrong. As mentioned in the introduction part of the documentation, this was our first project where we worked with a program language such as Python/Django. To summarize we are quite pleased on what we achieved, although none of us had any experience with coding but of course, we had our difficulties with the project. There are always things to improve and we hope we will get even better for the next web-based project.

## What went well?

* …1
* …2
* …3
* …4
* …5
* …6
* …7

## What went wrong?

* …1
* …2
* …3
* …4
* …5
* …6
* …7

List of references

Source Kumar Pal and Spiral Model picture : <https://www.geeksforgeeks.org/software-engineering-spiral-model/>

Sommerville 2011

Rosenberg and Stephen, 2007

Example using APA:

Fama, E., & French, K. (2015). Incremental variables and the investment opportunity set. *Journal of Financial Economics* 117, 470-488.

Swales, J. M., & Feak, C. B. (2009). *Abstracts and the writing of abstracts*. Ann Arbor, MI: University of Michigan Press.

Wooldridge, J. M. (2010). *Econometric Analysis of Cross Section and Panel Data (2nd ed.)*. Cambridge, MA: The MIT Press.

List of figures

The caption and numbering for a figure appears above or below the figure.

For references made in the text to figures, the corresponding number must be included. The reference “as the following figure illustrates” is not permitted.

In order to define the figure caption, use the respective function in Word.

Source references for figures are not part of the figure caption. They should be included on a separate line below.

Sources for the figures should not be included in the table of figures.

The type of figure caption must be identical to the type used for table captions.

List of tables

The caption and numbering for a table appears above or below the table.

For references made in the text to tables, the corresponding number must be included. The reference “as the following table illustrates” is not permitted.

In order to define the table caption, use the respective function in Word.

Source references for tables are not part of the table caption. They should be included on a separate line below.

Sources for the tables should not be included in the list of tables.

The type of table caption must be identical to the type used for figure captions.

List of abbreviations

ADL Arbeitsgemeinschaft für elektronische Datenverarbeitung und Lochkartentechnik e.V.

AGPLAN Schriftenreihe der Arbeitsgemeinschaft Planung e.V.

HBR Harvard Business Review

HdB Handwörterbuch der Betriebswirtschaft

RKW Rationalisierungs-Kuratorium der Deutschen Wirtschaft e.V.

REFA Verband für Arbeitsstudien e.V.

VDI Verein Deutscher Ingenieure

A list of abbreviations must be compiled where the paper contains uncommon abbreviations.

Example text for the first use of an abbreviation:

“The Rationalisierungs-Kuratorium der Deutschen Wirtschaft e.V. (RKW) recommends…”

Subsequently, the abbreviation “RKW” can be used in the text.

Abbreviations must be listed in (ascending) alphabetical order.

Common and widely understood abbreviations such as “e.g.”, “cf.”, “etc.” … should not be included in the list of abbreviations.

Annex

Any important information which is not crucial to understanding the statements and takes up substantial space should not be included directly in the text. Such information should be included in an annex at the end of the paper.

Interview transcripts, large amounts of data, etc. should not be included in an annex. They should be sent to your assessor electronically. Interview transcripts enable an assessor to understand the context in which a statement was made and to verify that the meaning of the statement has not been distorted.

Declaration of authorship

[Title - subtitle]

[I/We] hereby declare that the present paper is entirely my/our own work and without the use of any unauthorised assistance. Any content which has been taken verbatim or paraphrased from other sources has been identified as such. This paper has not been submitted in any form whatsoever to an examining body. Previously published work has been cited as such.

Vaduz, 16.12.2020

Alexander Büchel

Fabio Hilti

Lucy Gannon