**Tournament Bracket Generator**

Information Systems Development

Documentation

Alexander Büchel

*Matricula no.*

Fabio Hilti

*Matricula no.*

Lucy Gannon

*Matricula no.*

University of Liechtenstein

Master’s Program in Information Systems

Module: Information Systems Development

Assessor: Dr. Frank Breitinger

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

Date of submission: 17/12/2020

**Content**

[Introduction 3](#_Toc58252695)

[Project Description 3](#_Toc58252696)

[Project Goals / Requirements 3](#_Toc58252697)

[Project Group 3](#_Toc58252698)

[User Experience / Guideline 4](#_Toc58252699)

[Highlights 4](#_Toc58252700)

[Readme and Notes 4](#_Toc58252701)

[Conclusion 4](#_Toc58252702)

[Attachment 4](#_Toc58252703)

[Declaration of authorship 4](#_Toc58252704)

# Introduction

Short Introduction?

# Project Description

To be compete with one and each other is in the human nature – it is fun. 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 Bracket Generator” for our group project in the module “Information System Development”.

We plan to create a Bracket application with Phyton/Django in which a group of friends or a sport club can straightforward create a simple tournament bracket. There are some bracket generators on the word wide web, but most of them have unnecessary customization, which will take up a lot of time. That is why we decided to create a tournament generator without beating around the bush. Key elements of the projects are the three different tournament brackets, which will alter themselves on the registered results of the team.

Concept. Text what is the Project about? Short Introduction, show example of usages

Required Packages

# Project Goals and Requirements

The goal of this group project in the module “Information System Development” is to create a Tournament Bracket Generator, which – as its name already says – generate a bracket for tournaments with 8, 16 or 32 Teams. In Addition to the Bracket System the end-user of our application can create teams with team name, number of players, manager, and captain, which can be added to the created tournament.

The main target group for the usage of the “Tournament Bracket Generator” are mainly people, who want to organize a quick sporting event without big of a planning for sports such as football, tennis, handball and many more. Besides the that, the application is not only designed for sport events but rather for competitions, respectively competitive fun-tournaments in general (e.g. Gaming, Beer Pong, other Fun-Games).

With the created “Tournament Bracket Generator” our goal is to offer an application to create a Tournament Bracket, that can be used and implemented quickly in real life.

We have set following requirements at the beginning of our group project:

* Overall Greater understanding in Phyton and Django
* Creation of three different Tournament Bracket (8,16,32 teams)
* Random-Autofill for blank Fields for tournament and team creation
* Guided and simple process step by step (from tournament size to team creations to tournament bracket)
* Uniform and presentable Design for end users
* Clear Documentation of our group project
* (Nespresso) What else…?

What is the Goal of our Tournament Bracket Generator? Which Target groups are important? Who should benefit from this app?

# Project Group

Our project group consists of three people: Alexander Büchel, Fabio Hilti and Lucy Gannon. Each of us has little to none experience in Phyton, Django, Github and so on. For this reason, we had to gain our phyton and coding knowledge almost from scratch. Despite our limited experiences we successfully created our intended Django-project “Tournament Bracket Generator” successfully. Although each group member worked on every project task such as coding, documentation etc. at least for a little bit, we tried to split up the responsibilities.

Our communication was probably more difficult than the years before, due to the global pandemic, which hasn’t allowed us to meet in person for the project work. So, the communication for our “Tournament Bracket Generator” occurred online via WhatsApp or Zoom.

Alexander Büchel: Which Responsibilities? Experiences with Django, Background

Fabio Hilti: Which Responsibilities? Experiences with Django, Background

Lucy Gannon: Which Responsibilities? Experiences with Django, Background

The group consists of…? Who are we? What are our experiences with Phyton, Django…? What are our responsibilities in the group project?

# User Experience and Run through

Show with Pictures of our Generator, what we can do and how it works 🡪 step by step like a manual

# Highlights

Highlight the most important key functions of our project…

What are our Highlights or specialties in our project?

# Readme and Notes

A clear documentation of the used code in the “Tournament Bracket Generator” project is used with docstrings in the Django-project itself. In Addition, there is a Readme available in our shared GitHub.

Mention, that its code-descriptions/docstrings and readme are on github and/or on the python project itself

# Conclusion / In Hindsight

Retrospect: What went well? What was difficult? Are we satisfied with our final app? What would we change, if we had the possibility?

# Attachment

Everything that does not belong in the documentation paper

# Declaration of authorship

We here by declare that the present paper is entirely our own work and without the use of any unauthorized 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, 17.12.2020]

Alexander Büchel

Fabio Hilti

Lucy Gannon