Software Requirements Specification
- Connected.Football Vote4Fun -

Lucas Gehlen (2869144)
Marco Kull (3408140)
Patrick Richter (2478579)
Sebastian Wilczek (2746913)

January 13, 2019

Contents

1 Introduction

1.1 Purpose

This document was created in the context of the curse ”Software Factory” held at Fontys Venlo University of Applied Sciences in the winter semester 18/19. It provides a detailed description of our tasks to be a reference for development agreed upon by the product owner.

1.2 Scope

The Connected.Football app is a smartphone application for managing the training of football teams. Our task is to extend the already existing app with more functionality, introducing a feature to allow the players to take part in the planning of the training and therefore enable them to get more involved in the training.
To achieve this goal, the Connected.Football app should allow a coach to provide her / his team players with the opportunity to vote for their desired exercise for scheduled training sessions.

1.3 Definitions, acronyms, and abbreviations

Term	Definition
coach	A user with the role ’coach’ assigned within a particular club, assigned to one or more teams of that club.
player	A user with the role ’player’ assigned within a particular club, assigned to a particular team of that club.
user	A person registered to the Connected.Football application.

1.4 References

This document is based on IEEE standard 830-1998 - ”Recommended Practice for Software Requirements Specifications”.

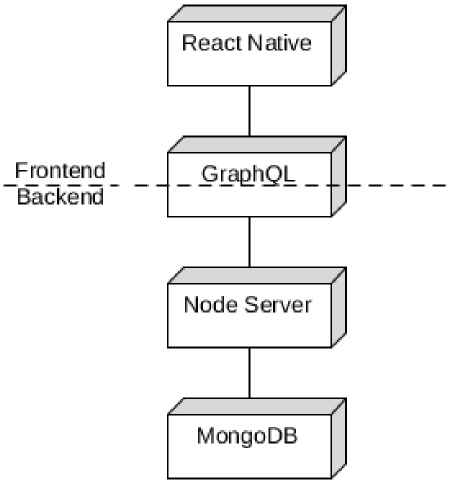
1.5 Overview

The rest of this document is divided in two chapters. Chapter two provides an overview of the systems functionality, its interaction with other systems and stakeholders, as well as some constraints and assumptions about it. The third chapter explains the requirements specification in detail using different specification techniques.

2 Overall Description

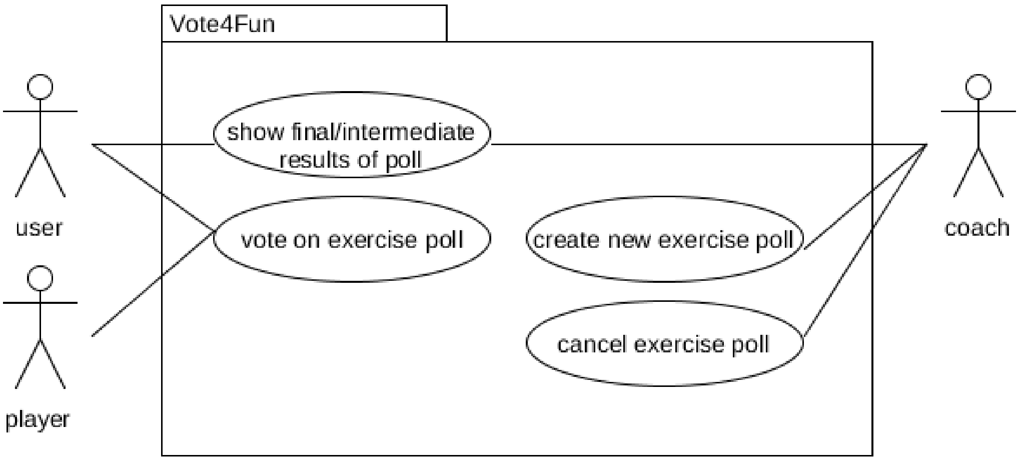
This section is intended to give an overview of the whole system and explains its interaction with other systems, its basic functionality and the stakeholders from whom it will be used. Furthermore some constrains and assumptions are documented.

2.1 Product Perspective



The applications' frontend is based on *React Native*, it uses *GraphQL* to communicate with a node server saving its data in a *MongoDB* database.

2.2 Product Functions



2.2.1 User

Every person that is registered at the Connected.Football application is considered as a user. A user has one specific role that is assigned by the customer.

2.2.2 Coach

A coach is a user with the specific role of being a coach. As such she or he is assigned to a specific club by the customer and can have multiple teams he is responsible for.

2.2.3 Player

A player is a user with the specific role of being a player. As such she or he is assigned to a specific club and team.

2.3 User Characteristics

There are two types of users involved using the Vote4Fun feature: a coach and players. Both types of users are heterogeneous groups with a huge variety of technical skills. Most of the people using the feature will want to have a quick and easy way to create and take part in the poll or they'll skip it.

In particular the coaches should be thought of, because the creation of the polls is a crucial part of the feature. Thus it should be easily accessible as too much complexity may distract them.

3 Specific Requirements

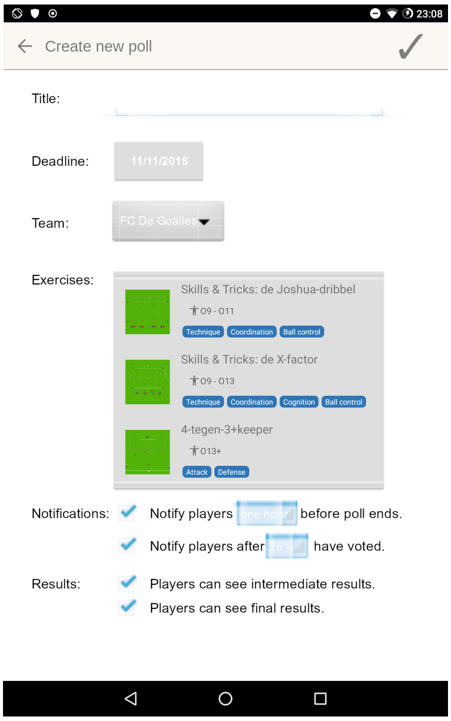
This section contains all of the functional and quality requirements, giving a detailed description of the Vote4Fun features.

3.1 External interface requirements

This section provides basic prototypes of the user interface. It also describes all inputs and outputs from the Vote4Fun module.

3.1.1 User interfaces

After creating a training a coach should be able to add a poll for it, allowing to set custom preferences:



23:08

Create new poll ✓

Title:

Deadline: 11/11/2018

Team: FC De Goalies

Exercises:

Skills & Tricks: de Joshua-dribbel
O9 - O11
Technique Coordination Ball control

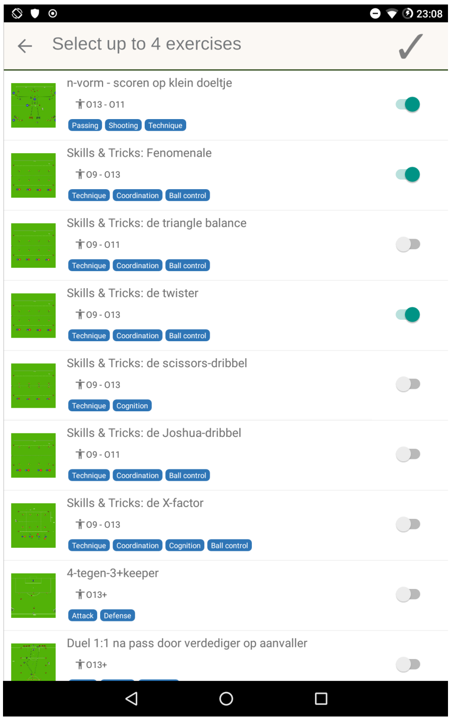
Skills & Tricks: de X-factor
O9 - O13
Technique Coordination Cognition Ball control

4-tegen-3+keeper
O13+
Attack Defense

Notifications: ☑ Notify players one hour before poll ends.
☑ Notify players after 75% have voted.

Results: ☑ Players can see intermediate results.
☑ Players can see final results.

Part of this creation process is to select the desired exercises that will be voted upon:

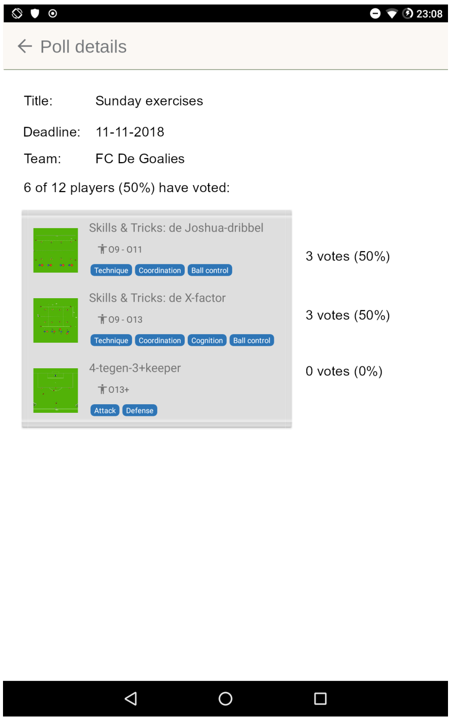


23:08

Select up to 4 exercises ✓

n-vorm - scoren op klein doeltje O13 - O11 Passing Shooting Technique	●
Skills & Tricks: Fenomenale O9 - O13 Technique Coordination Ball control	●
Skills & Tricks: de triangle balance O9 - O11 Technique Coordination Ball control	○
Skills & Tricks: de twister O9 - O13 Technique Coordination Ball control	●
Skills & Tricks: de scissors-dribbel O9 - O13 Technique Cognition	○
Skills & Tricks: de Joshua-dribbel O9 - O11 Technique Coordination Ball control	○
Skills & Tricks: de X-factor O9 - O13 Technique Coordination Cognition Ball control	○
4-tegen-3+keeper O13+ Attack Defense	○
Duel 1:1 na pass door verdediger op aanvaller O13+	○

After creation a dialog is integrated that shows the main information, and dependent if the active user already voted - either vote buttons or the percentage of users that voted:



Title: Sunday exercises

Deadline: 11-11-2018

Team: FC De Goalies

6 of 12 players (50%) have voted:

Skills & Tricks: de Joshua-dribbel O9 - O11 Technique Coordination Ball control	3 votes (50%)
Skills & Tricks: de X-factor O9 - O13 Technique Coordination Cognition Ball control	3 votes (50%)
4-tegen-3+keeper O13+ Attack Defense	0 votes (0%)

3.1.2 Software interfaces

To be able to handle a poll data must be stored, but access to the code was only given to the frontend. Together with the customer we discussed the data we would need and came up with the following representation in *JSON*:

```
{
    id: {
      type: SimpleSchema.RegEx.Id,
      optional: true
    },
    title: {
      type: String,
      min: 1,
      max: 50
    },
    description: {
      type: String,
      max: 10000,
      optional: true
    },
    trainingTarget: {
      type: String,
      max: 200,
      optional: true
    },
    exercises: {
      type: Array
    },
    'exercises.$': {
      type: Object,
      optional: true
    },
    'exercises.$.referenceId': {
      type: String,
      optional: SimpleSchema.RegEx.Id
    },
    'exercises.$.duration': {
      type: Number,
      min: 0,
      max: 30
    },
    'exercises.$.comment': {
      type: String,
      optional: true,
      max: 1000
    },
    'exercises.$.trainingPhase': {
      type: String,
      optional: true,
      max: 50
    },
    fieldSize: {
      type: Number,
      allowedValues: [125, 250, 500, 750, 1000]
    },
    nrOfPlayers: {
      type: Number,
      optional: true,
      min: 0,
      max: 20
    },
    trainingDate: {
      type: String,
      optional: true,
      // https://stackoverflow.com/questions/3143070/javascript-regex-iso-datetime
      regEx: /(\d{4}-[01]\d-[0-3]\dT[0-2]\d:[0-5]\d:[0-5]
         \d\.\d+([+-][0-2]\d:[0-5]\d|Z))|(\d{4}-[01]\d-[0
         -3]\dT[0-2]\d:[0-5]\d:[0-5]\d([+-][0-2]\d:[0-5]\
         d|Z))|(\d{4}-[01]\d-[0-3]\dT[0-2]\d:[0-5]\d([+-]
         [0-2]\d:[0-5]\d|Z))/
    },
    teamSharing: {
      type: Object,
      optional: true,
    },
    'teamSharing.teamId': {
      type: String,
      regEx: SimpleSchema.RegEx.id,
    },
    'teamSharing.coach': {
      type: Boolean,
      defaultValue: false,
    },
    'teamSharing.coordinator': {
      type: Boolean,
      defaultValue: false,
    },
    'teamSharing.player': {
      type: Boolean,
      defaultValue: false,
    },
    'teamSharing.vote4fun': {
      type: Boolean,
      defaultValue: false,
    },
    vote4fun: {
      type: Object,
      optional: true,
    },
    'vote4fun.type': {
      type: String,
      allowedValues: ['clubteam', 'open'],
    },
    'vote4fun.title': {
      type: String,
      min: 3,
      max: 40,
    },
    'vote4fun.exercises': {
      type: Array,
        minCount: 2,
        maxCount: 4,
      },
      'vote4fun.exercises.$': {
        type: Object,
      },
      'vote4fun.exercises.$.exerciseId': {
        type: String,
        regEx: SimpleSchema.RegEx.Id,
      },
      'vote4fun.exercises.$.playerIds': {
        type: Array,
      },
      'vote4fun.exercises.$.playerIds.$': {
        type: String,
        regEx: SimpleSchema.RegEx.Id,
      },
      'vote4fun.deadline': {
        type: String,
        // https://stackoverflow.com/questions/3143070/javascript-regex-iso-datetime
        regEx: /(\d{4}-[01]\d-[0-3]\dT[0-2]\d:[0-5]\d:[0-5]\d\.\d+([+-][0-2]\d:[0-5]\d|Z))|(\d{4}-[01]\d-[0-3]\dT[0-2]\d:[0-5]\d:[0-5]\d([+-][0-2]\d:[0-5]\d|Z))|(\d{4}-[01]\d-[0-3]\dT[0-2]\d:[0-5]\d([+-][0-2]\d:[0-5]\d|Z))/,
      },
      'vote4fun.guestIDs': {
        type: Array,
        optional: true
      },
      'vote4fun.guestIDs.$': {
        type: String,
        regEx: SimpleSchema.RegEx.Id,
      },
      'vote4fun.showIntermediaResults': {
        type: Boolean,
        defaultValue: false,
      },
      'vote4fun.showFinalResults': {
        type: Boolean,
        defaultValue: false,
      },
      'vote4fun.notificationHoursBeforeDeadline': {
        type: Number,
        defaultValue: 1,
        min: 1,
        max: 24,
      },
      'vote4fun.notificationAfterPercentageVoted': {
        type: Number,
        min: 0,
        max: 100,
      }
}
```

Listing 1: *Vote4Fun JSON Object*

The data endpoint was provided by the customer. The idea, besides the self-explaining preference values, is that for every selectable exercise there is an array with user ids. When a user votes for an exercise her or his user id will be saved in that specific array. This way it is always possible to know which user voted for which exercise - and is also open for features like removing a already placed vote.

3.2 Functional requirements

3.2.1 Create New Exercise Poll

Use Case	Create New Exercise Poll
Stakeholder	coach
Precondition:	- A coach user is signed in to Connected.Football app. - Actor has opened the training overview page.
Flow of events:	1. Actor selects create new exercise poll. 2. System asks for a poll title. 3. Actor enters a title for the poll. 4. System shows a list of available teams. 5. Actor pick a team from the list. 6. System shows a date picker to select the deadline of the poll. 7. Actor selects a time using the date picker. 8. System shows a list of exercises that could be voted upon. 9. Actor selects at least 2 exercises but max 4 for the poll. 10. System shows a dialog with all information entered. 11. Actor presses accept button. 12. System returns to the edit training page. 13. Actor presses save button.
Exceptions:	- The training already has a poll assigned. - The training date is in the past. - The poll title should not be empty. - The team should not be empty. - The deadline of the poll should be in the future. - The deadline of the poll should end before the training starts. - The selected poll exercises should not contain copies. - The selected poll exercises should not already be part of the training.
Post condition:	The system created a poll for the selected training.

3.2.2 Cancel Exercise Poll

Use Case	Cancel Exercise Poll
Stakeholder	coach
Precondition:	- A coach user is signed in to Connected.Football app. - A poll was previously created for the selected training. - Actor has opened the poll overview page.
Flow of events:	1. Actor presses the delete button. 2. System asks for confirmation. 3. Actor presses the confirm button.
Exceptions:	- The training date is in the past.
Post condition:	The system deleted the poll for the selected training.

3.2.3 Vote On Exercise Poll

Use Case	Vote On Exercise Poll
Stakeholder	player
Precondition:	- A player is signed in to Connected.Football application. - Actor is part of a training for which a poll was created. - Actor navigated to the poll overview page.
Flow of events:	1. System displays the available exercises that can be voted upon. 2. Actor presses the vote button alongside the desired exercise. 3. System shows a confirmation message.
Exceptions:	- The deadline is in the past.
Post condition:	The system saves the player's choice of exercise.

3.2.4 Show Final/Intermediate Results Of Poll

Use Case	Show Final/Intermediate Results Of Poll
Stakeholder	user
Precondition:	- Actor is signed in to Connected.Football application. - Actor is part of a training for which a poll was created. - Actor is either the coach or a player that already voted. - Actor navigated to the poll overview page.
Flow of events:	1. System displays the available exercises with percentage of votes.
Exceptions:	- Actor is a player and poll does not allow showing results (yet).
Post condition:	-

3.3 Design constraints

The appearance of the Vote4Fun module should integrate into the layout of the Connected.Football application.