TEAMFINDER

Your Name <beki18yh@student.ju.se>
Your Name <axax18ms@student.ju.se>

A Project Work in *Web Development – Advanced Concepts*

Jönköping University 2020

In this template, all italic text should be removed and replaced with your own text (which should not be italic); the italic text is just a placeholder letting you know what to write there.

On the cover page, change to your own platform name, your own name and your own JU email address.

You have a lot of freedom when it comes to writing this report. You do not have to use any part of this template, but the report you write should in the end somehow (in a good way) provide the same information as indicated in this template. Most students trying to do it in their own way usually fail, so if you try that, be sure to know what you are doing!

This page should of course be removed.

Table of Contents

Introduction

*Introduce your platform here. Write text that **indirectly** answers questions like:*

- *Why was the platform created?*
- *What can the platform be used for?*
- *Why would anyone use the platform?*
- *Who will be using the platform?*
- *How will users use the platform?*
- *…*

A common problem in society is to meet new people in a fun and interactive way. A simple solution to this is to invite people who are similar to yourself in terms of age and hobbies who you can participate in activities with. That is easy in theory, but it might be hard finding those kinds of people. That is where this platform comes in. It is designed to help people in all ages to find new friends and other like-minded people to play sports with!

That is implemented by letting a user create groups with information about what sport they want to play, age limit, and in which city they play in. The post will become public to all users and if the requirements are met by a user, they can join the group and start talking to the other group members using a text chat within the group. In order to become a user, you need to register an account on the website. Before that you're a guest and can only see a limited amount of content on the platform. More about that in the architecture section.

Everything within this area is part of the mockup and not something that will be found in the report later on. You can find all of the below in the image in the Database section.

Resources:

- *Activities – When joining an activity, you connect to the group chat*
- *Group chats – Used to communicate and set up a schedule*
- *Accounts*

Accounts:

- *Account id*
- *fnamn*

- *Lnamn*
- *Email*
- *Age,*
- *Telefon, utfyllnad*
- *City, var man är*
- *Kön, utfyllnad*

**smånga till amband* GroupMembers:*

- *Account id (foreign key)*
- *Grupp id (foreign key)*

Explanation of connection, one account can exist in several groupMembers entites. One GroupMember Entity can only have one account connected to it.

Group:

- *Group id*
- *Name, för o identifiera grupper för användaren*
- *Nr of members, hur många som finns med*
- *MaxNrOfMembers, hur många medlemmar som får va med*
- *(City)*
- *(Age limit, xx-xx, krav för att gå med I gruppen)*
- *(Tillåtet kön, krav för att gå med)*
- *(datum, om man skapar gruppen för ett speciellt datum... osäker på att vi gör så)*

Message:

- *Message id*
- *Group id (foreign key, group)*
- *Text*
- *Datum vid skapande*
- *Account id (foreign key, group) , behövs om meddelande ska länkas till account o ett account ska kunna ta bort/redigera meddelande*
- *AuthorName, sparas från kontot vid skapande av meddelande*

Add a UML use case diagrams that visualizes how end users will use the platform.

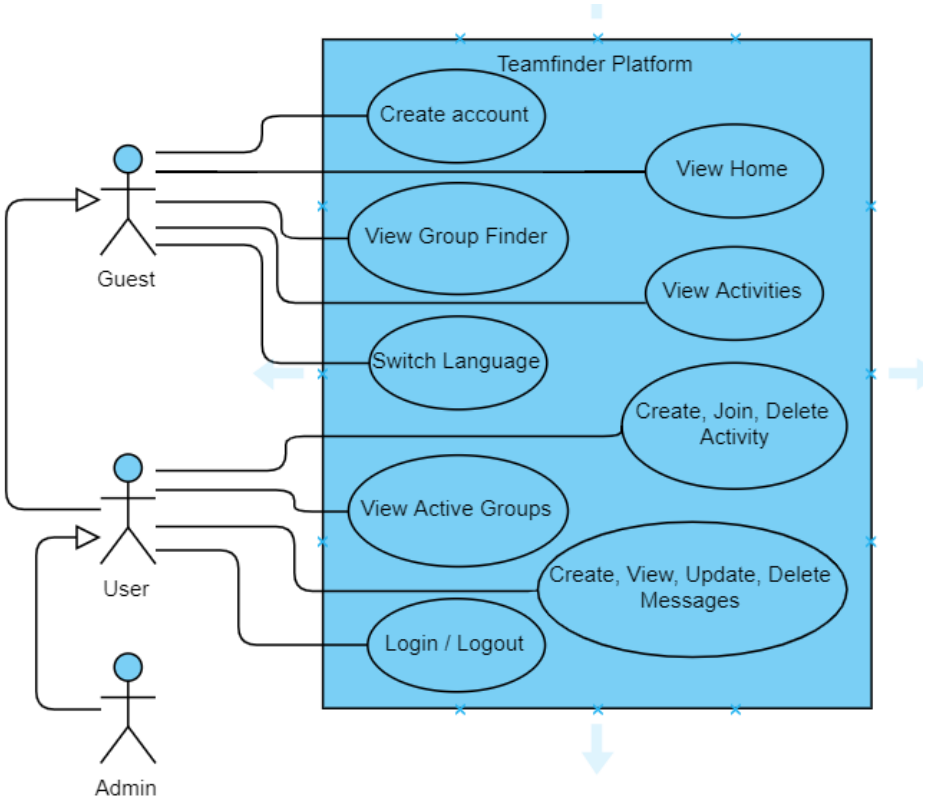
After having read this chapter, those that have never heard of the platform before should a have a good understanding of what it is about. If they would like to learn how it has been implemented, they just need to continue reading the rest of the report.

Architecture

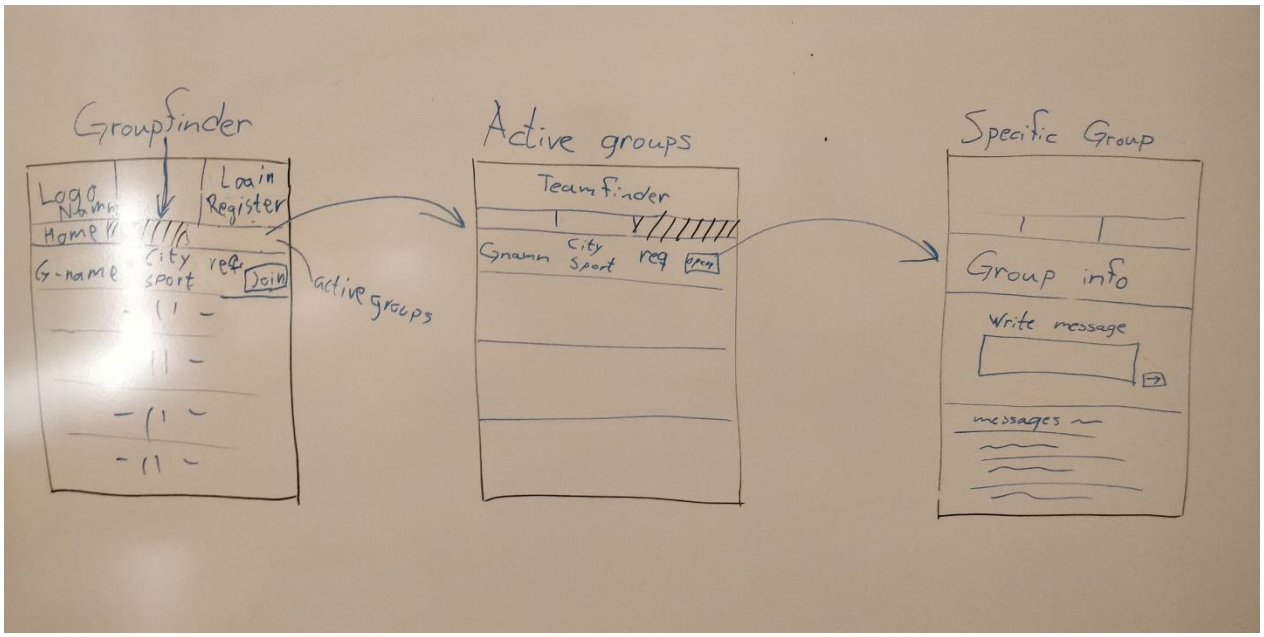
Give an overview of the components the platform consists of (web application, database, web browsers, SPA, end-users, etc.). Visualize this using a figure and show how the different components make use of each other.

Are you using Docker? Then it also makes sense to describe which parts of the platform that run in which containers in this chapter. Maybe visualizing this with a figure is a good idea? ***Hint: Yes, it is****.*

After having read this chapter, the reader should have a broad (but shallow) understanding of the platforms internal components and structure.



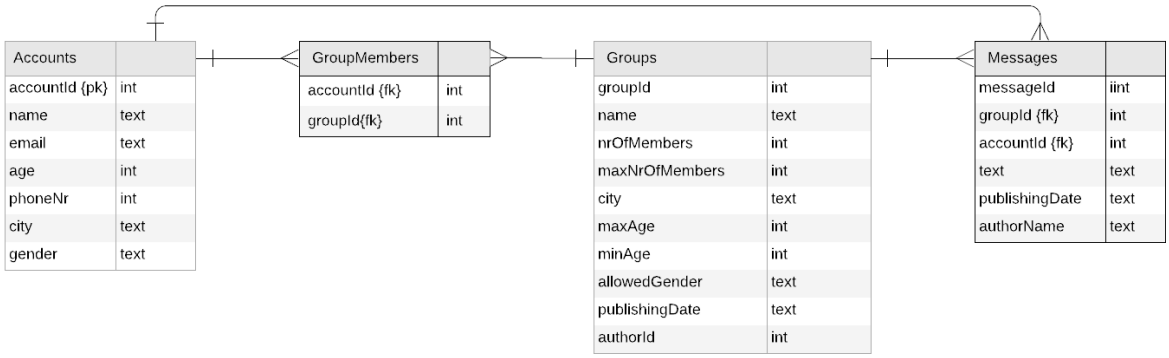
The header UI will be based on the following website: https://www.teamfinder.gg/


Groupfinder
Logo
Login
Register
Home
City
sport
req
G-name
Join
active groups
Active groups
Teamfinder
City
Sport
req
Specific Group
Group info
Write message
messages

Database

Describe the resources on the platform in detail. What attributes do they consist of? How are they related? Maybe an ER diagram is suitable to have? ***Hint: Yes, it is.***

After having read this chapter, the reader should understand how the data on the platform is stored and structured. If the reader is a new programmer that should start working on the website, she should now know what she needs to do if she wants to change a resource or add more resources (e.g. know how to add a new table to the database with a relation to an existing table in the database, etc.).



GroupMembers:

AccountId + GroupId = candidatnyckel?

Web Application

Describe implementation details of the web application. Which language have you used? Which framework have you used? Which libraries/packages have you used, and for what purpose?

Has all code been written in one file? Or have you somehow structured it in multiple files? Or layers? Are you using some design patterns (e.g. MVC)? Are you using middlewares? Etc... Maybe visualizing this using figures is a good idea? ***Hint: Yes, it is.***

You do not necessarily need to show any code to describe the implementation, but if you feel that improves the quality of the report, feel free to do that.

Does your web application expose a REST API? Then you should also specify how others can use the resources on the application through the REST API. Mention all details about the REST API others needs to know to use it. Maybe describing that in another main chapter is a good idea?

After having read this chapter, the reader should have a very good understanding of how the web application has been implemented and how the code is structured. If the reader is a programmer who should start working on the web application, she should now know where to start when she should implement new features to the web application.

Single-Page Application

Describe implementation details of the Single-Page Application.