

SYSTEM SPECIFICATIONS

MINDFIRE

Andreas Reischl, Marko Diljkan, Egon Manya



# 1 Initial Situation and Goal

## 1.1 Initial Situation

There are a lot of other quiz apps which have some very good features and implementations. Two of the most popular quiz-apps are Quizduell and QuizUp, which are available for Android and Iphone.

In Quizduell you are able to play online quizzes, but only against one person and it is not real-time but turn-based. Others than Quizduell, our app will have a real-time multiplayer and the users are going to be able to play against more than one player. There is free a free and premium version of Quizduell. In the free version you are only able to play and in the premium version you also can create questions, access your personal statistics and much more.

The QuizUp is for free and it also has a real-time multiplayer. Like mostly every quiz-game in the app-store, QuizUp only supports 1vs1 sessions. The questions are categorized and you are able to create own questions or categories, but only over the webapp of QuizUp and not over the mobile phone.

Another quiz-app, which got very popular over the last time, is Kahoot. It offers a real-time multiplayer and it’s playable on Android, Iphone and even on the Webapp, but it’s only for private sessions and doesn’t offer many other functions.

Also there is no app which features guessing questions. In most of the quiz-apps there are only questions, where you have the choice between four different answers. Our app brings the new guessing playmode to the traditional quiz playmode.

## 1.1.1 Application Domain

The probably most important part of the application domain is that our app can really help people learn new things and can be used by pupils to pass through exams. The user can easily create a category, add questions and compete with colleges. That has the effect that the pupils learn things effortlessly by playing Mindfire against each other.

Another great part of the application domain is that teachers can use the app to brighten up the lessons at school and strengthen the knowledge of students by short quizzes. It also brings the possibility to check how good the pupil’s knowledge currently is by the result of the quizzes.

The last part of the application domain is that it helps passing time away and expanding the player’s knowledge.

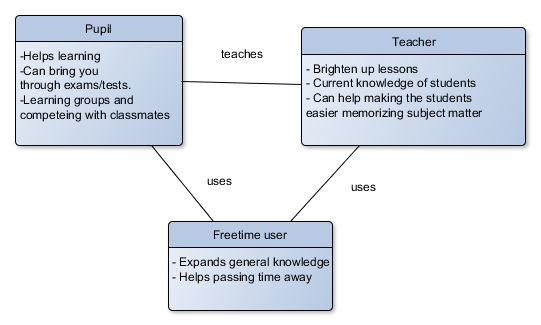
**1.1.2 Glossary**

<Learning><Helps learning new things. Can be used a preparation for tests or exams. Own question and categories. Competing with other students>

<Teaching><To brighten up lessons. Having a view of the student’s current knowledge. Helps teaching students>

<General knowledge><Helps passing time and additionally focuses on expanding the general knowledge of the user>

**1.1.3 Model of the Application Domain**



**1.2 Goal Definition**

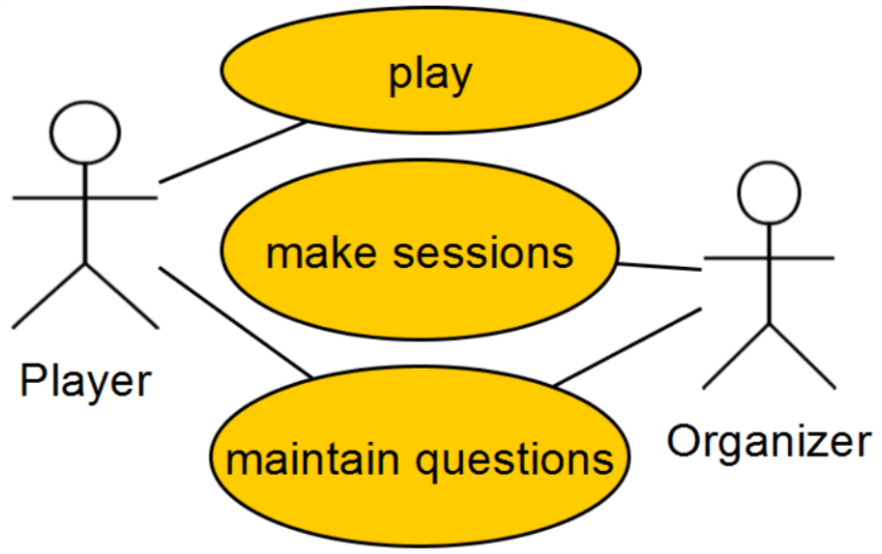
The main goal of our app is to ensure a real-time multiplayer quiz game where it’s possible to play against more than just two players at once. There are two kinds of questions. Those where the user has to estimate a value to a certain question or picture and those where one out of four answers to question or picture has to be guessed. To make the quiz more interesting there’s a timer, which means that the faster you answer the higher the score you get for answering. For answering estimation questions are also score points. The closer the answer is to the real result the more score points are gained. When answering a question with given answering options the answer picked must be correct to get time bonus score points. We also want to ensure that the user can create his own questions and categories and even make them public. The user can play private sessions with friends or people invited and with any public players.

Our target group is nearly everyone. But especially we are focused on young people, who like to try new apps and like quizzes. Also people that simply want to mess with other in different categories of knowledge and their general knowledge. Even those that just want to learn new things are important. For us it’s also important to have a target group that wants to use the app to learn for exams or tests and have a really helping tool with this, because we think that this could be a very funny and cool new way to learn. Another target group are people, that have a teaching or instructive role and want to use the app as a resource for teaching. The last group are those we want to satisfy for our app and that do not really have any idea of quiz games and that aren’t really convinced why to use them yet.

Concerning prerequisites, we can say that aren’t really any required. There might be some benefits for users that already played quiz games or are very interested in certain categories of the game, because they just have a big advantage competing against others. But to simply use our app there aren’t any prerequisites required.

# 2 Functional Requirements

**2.1 Use Case Diagrams**



**2.2.1 Characteristic Information**

Use Case 1: play

|  |  |
| --- | --- |
| Goal: | scoring more points than others |
| Precondition: | create or connect to a session |
| Postcondition: | leaderboard and scores will be shown |
| Involved User: | player |
| Triggering Event: |  |

Standard use

As the player enters a game, the player will get the defined number of question to answer. The player will answer the questions in series and for every questions we have particular time to respond. As quicker you answer the right question the more points the player are going to get, but if the player doesn’t answer in time or choose the wrong answer, he won’t get any points. After every player finished the game, the player is going to see a leaderboard with all players ordered by points.

Non standard use

Connection

The first thing before using our app is to ensure that a connection to the internet is given. The app won’t work without an internet connection. This is because it’s not possible to play an online guess quiz other players at the same time without being connected with each other. But also things like statistics or user date won’t be able to be seen or accessed without connection to the internet. All the date is stored on a server so it won’t be accessible. The only problem with this issue is that we don’t really know in how far the app will be inflicted by internet issues. One solution would be to check if the user can connect to the server and is connected to the internet and in case that this is not given the app can’t be used. Another solution would be to have a few features like it’s possible to write questions for the quiz and categories which will be saved locally as drafts and can be uploaded to the server when the mobile phone goes online or that the user can train the quizzes in alone and tries to score as many points as possible in a solo mode.

Disconnect

Another issue that could come up is when an online multiplayer game a player loses the connection to the game and server. A common scenario would be just to continue the game session, which means that the others players which are still connected can continue playing without the missing player. We also thought about the opportunity to let the other players which are still connected vote to wait for the disconnected player or to continue the game. If the missing player doesn’t join the game in a time like 30 seconds or something the game continues without him. Or that the possibility of rejoin the game exists. This means that all the others players continue playing if one player disconnects and the missing player has the chance to rejoin when there’s a connection available again. But this is practically a little bit disadvantageous for the rejoining player, because the game continues while he’s trying to reconnect and his opponents can score the time he’s gone. So he woT

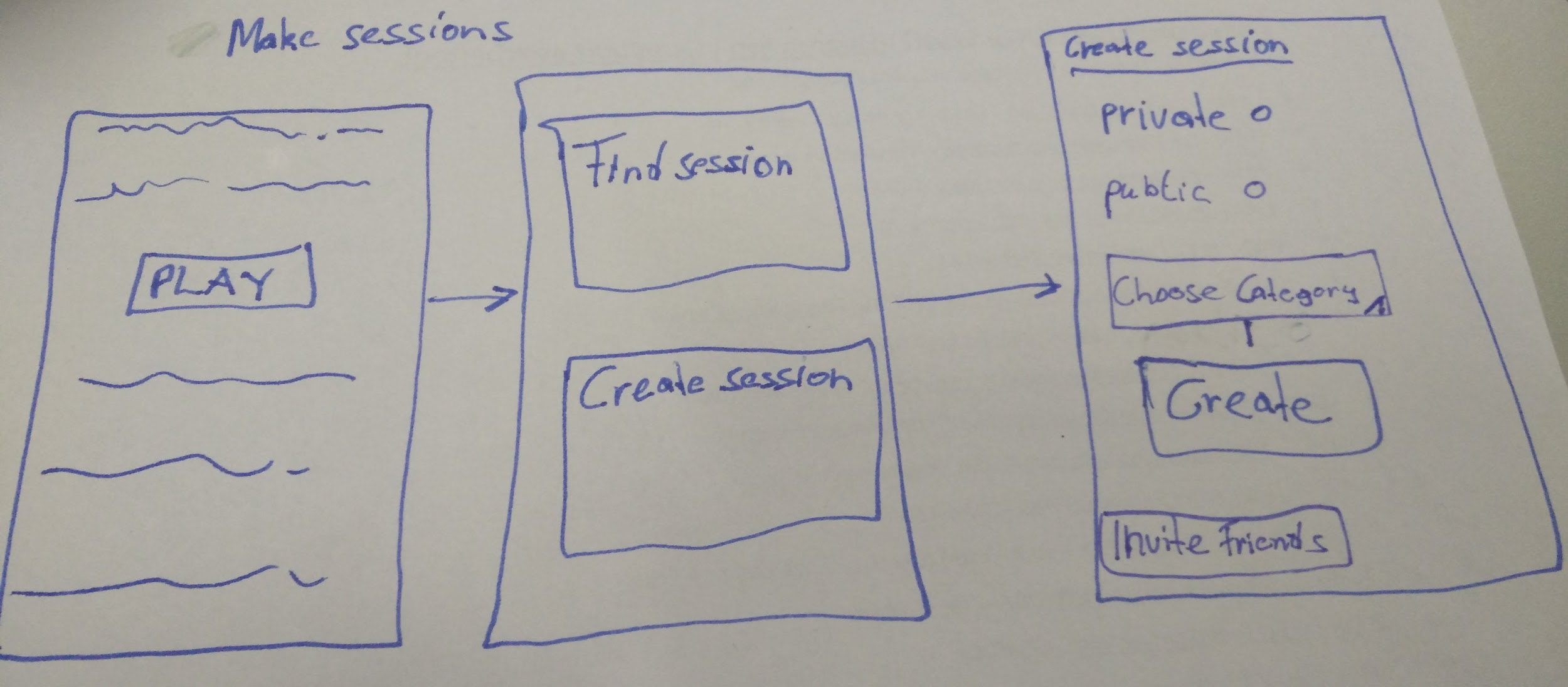
n’t probably catch up with the other player’s store and for sure not win anymore. The last option is to quit the session and cancel the current game if one player leaves or disconnects, but that will rather not be implemented or rather happen by a vote system. So the players are voting if the game should end or go on. When there are only two players playing a match and one of them disconnects it might be an opportunity to quit the game, because then there’s only one player left and if this player doesn’t want to wait the game could be quit. Our objective is to create the best variation of all this opportunities.

Open Point: Private Session hosting

We thought a lot about the server and if it will be able to handle a lot of private sessions with many players and had the idea that maybe it would be good (for our server) if the user that created a private session can also host it. Which means that his mobile phone/computer will handle the game. So we will have a lot more power on our server for the non-private game sessions. A school class with 30 pupils which would start a private session would take a lot of resources from the server, if such a session can be hosted by for instance a teacher or any pupil this would really help saving server resources. The only problem is that if the session host is for instance in a train and drives through a tunnel the connection might be lost and the game is without connection. In this case there would also be the possibility to have a host migration and someone else will be picked to take over the host part.

Use Case 2: make sessions

|  |  |
| --- | --- |
| Goal: | make a session for public or private to start the game. it should be able to invite friends |
| Precondition: | button click on “Create session” |
| Postcondition: | session successfully created |
| Involved User: | Every user who uses the app |
| Triggering Event: | When the user hits a button |



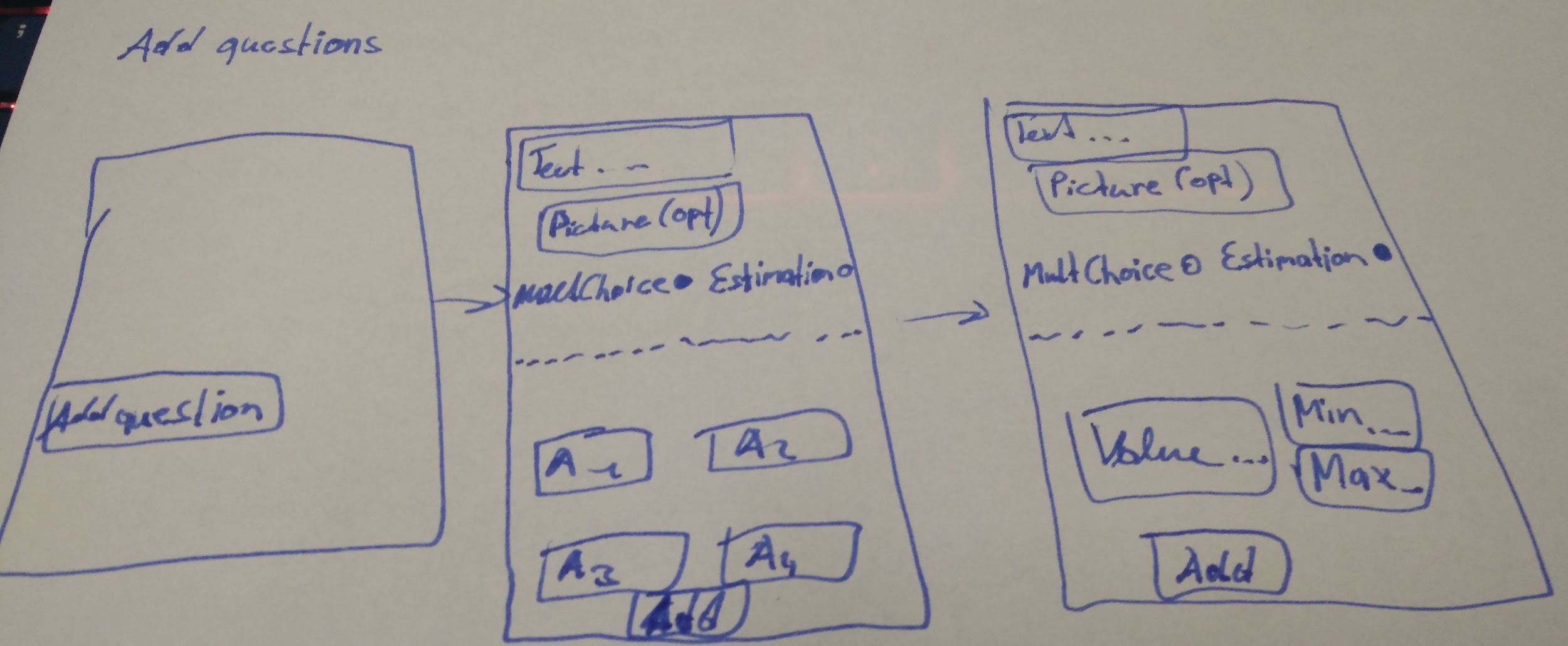
To make a session the user has to hit the “Create session” button. Than it can be chosen which type of a session should be created. Afterwards the user can select questions by categories and at least as an optional feature it will be available to invite their friends from Facebook.

Non-standard uses

When the user creates a new session it is required to add minimum 1 question and a name for the session. If the validation fails the user will get a detailed error log. The user might not have any friends online and a session can’t really be created.

Use Case 3: maintain questions

|  |  |
| --- | --- |
| Goal: | managing questions in the community |
| Precondition: | questions should be meaningful and it should not exist in database |
| Postcondition: | questions successfully added by category |
| Involved User: | The user who want to add questions |
| Triggering Event: | When the user hits a button “Add Questions” |



The GUI preview upon should show how a single question will be added. Of course there isn’t a category selection shown yet. We came to the conclusion that it’s not really necessary to have a draft of how to add questions to a certain category, because there will be the option the choose a category beside the “Add question” button where you will be directed to the “Get question pool” Use Case. However, as shown above a text can be chosen and optionally even a picture in addition to the question. Then the user can either pick “Multiple Choice” or “Estimation”. When “Mult Choice” was picked, four answers can be created and if “Estimation” is picked, the user has to enter the correct value, the minimum value and the max value of possible numbers.

Non-standard uses

To add question the user has to write a question and choose a question type. Than if a multiple choice question is selected the user has to give 4 different answers and select one correct answer. For an estimation question the user has to give a number as an answer. The user must give a min, a max and at least the exact value. If the validation fails the user will get a detailed error log.

One problem comes up by letting the users add questions, because it’s never sure that the questions added make sense or that they are really able to be quiz played by many other players. So we thought about giving the community the opportunity to vote the questions. It must be possible to up-vote and down-vote the question or the report them. But it’s never 100% sure if this system will really work. If it doesn’t work it will be necessary to check the question and remove or change them. In case, that there will be pictures for the questions, we are not yet completely sure, if the user may upload lots of pictures or even any pictures, because this could cause very high memory usage on the server. Either we limit the amount of pictures or memory the user may upload or we don’t let the user have this opportunity and limit the pictures and questions for pictures ourselves. This is because images can use a lot of memory.

**Open Point: Category by users**

Categories also count to maintain question and with these also comes an open point from our side. We are not yet 100% sure if it’s a good idea to let the user add their own categories, because when there is a category like “Food in Austria” and then someone adds a category “What Austrians eat” or for the “Food in Austria” category has only 10 questions and there is a category “Food in Europe” which has 40 questions it would not be very good to have an own category for these 10 questions. So instead they should be included to the “Food in Europe” category. As it can be seen clearly people might add the same categories just with a little different naming. On the other hand everyone could add categories and the people could build really awesome questions to categories that we (the developers) would not have created. We planned to give these feature to the users as a kind of testing and if it works we can keep it.

**3 Non-functional requirements**

|  |  |  |
| --- | --- | --- |
| Type | Name | Description |
| USE | Usability requirement | This requirement is to make the target group as described in section 1 is liking to work with that system. |
| EFFIC | Efficiency requirement | Run-time and memory efficiency. What are the constraints under which the system has to run. |
| MAINT | Maintenance and portability requirement | Which maintenance or porting effort is expected in the future? Internationalization expected? Porting to different hardware platform?... |
| SEC | Security requirement | Security requirements comprise confidentiality, data integrity, and availability. How much do we have to consider that data is not accessible to unauthorized persons? Is the correctness and/or consistency of data to be guaranteed? How severe are total system faults? |
| LEGAL | Legal requirement | Are there any standards or legal constraints to be considered? |

Usability

To make our app most useable for all our target groups there is a combination of a simple user interface and a very feature filled user interface. The app comes with a lot of features the user should be able to use like creating categories, adding questions, vote questions, create game session with different options, etc. but it still should be very user friendly, because nobody wants to use an app with big user interface complexity. While users that are using the app a lot and almost all the features it comes with might want a lot of possibilities and maybe even a user interface which brings a lot of content, casual users are not going to be very happy with so much content on the user interface. We want to hold a certain structure. The whole screen of the mobile phones will be used to bring medium content which is very clear and fast to use. The idea is to sort the big topics of our app into smaller topics with options that can also have topics, so that the users doesn’t have to click or chose a lot of options to get the relevant usage out of the app. For instance, it would be very annoying to have all categories listed on the home screen or even a few. We want to make the categories an own rubric on the title screen. The fastest thing should be to play the game, so we want a big great button where you can click and play without a lot of effort, but though there can be put a lot of features in it. This could look like you click a big button the the text “Play now!” on it and there will directly a search for other players online. While players are searched there’s the possibility to play a private game, because there’s a play private button, that can optionally be pressed during the searching process and the user can then invite friends and play private with them. Or beside the “Play now” button there is the possibility to start a private game instant before searching. For users that want to play with their friends there can also be a way to have group of friends before even clicking the “Play now” button and when a user is in such a group and the leader clicks “Play now” everyone in the group will stick with their friends in the group and search for others players online. What we want to express with these instances is that we bring a lot of ways to actually do the same thing with on the one hand more or less options and features and on the other hand with more or less effort according to the users’ preferences. People who use the app just as an additional teaching function will have an easy way to do this, because the user interface should be capable for this certain target group. Casuals will have it very easy and fast to play in a very short time with the least effort. Solo players that never play games with friends have a fast way to always just play and friends or people who like to play in groups can easily create their own group and join other groups or players within their group fast. And also those who want to pass the next week’s exam or test should be able to fastly access their category with all the question concerning learning matter and do their learning with other players. To maintain own questions and categories our web solution will bring a very comfortable way this. Creating questions on a keyboard is (for the most target groups) just more pleasant than typing them on a smartphone so even this is very fast and easy to handle.

Efficiency

As our app is a real time multiplayer app where it’s about fast reactions and even score points to give answers as fast possible it’s very important to ensure that everything runs fluently and that the communication between server and the participants smartphones goes along smooth. When a player answers a question in a quiz round the answer must be transmitted as fast possible in the range of a few milliseconds. If it takes a half second until the answer arrives the server it’s just too slow. This can’t be handled 100%, because the strength of the user’s internet connection is very relevant, but we try to make the server handle the answers as fast as they arrive. We don’t really want a delay. Also, we thought about to respect the reaction time of every user’s smartphone. If a player currently has a ping time of 300ms we can subtract these 300ms from the time the server has got to get the real result. This might only be necessary, when two players have a really similar time or one player has a really high ping. A quiz round only starts when every player is ready and loaded so that the server and player’s smartphones are really focused on transmitting the answering times to measure who answered first.

Also the search for other players should be efficient. It doesn’t really make any fun to wait endlessly for other players. So it should be possible to play a game with only two players already up to the maximum amount of players. Eight players is good number of players to play a public online game, of course players can create a game session where it’s possible to have more participants like a group of 30 people. Back to the search game, a “Ready” button will make the players search more comfortable. So when a time of 30 (maybe some more or less seconds) seconds has run out and only 3,4 or maybe 6 players are found everyone can play ready and the game starts without searching for further players. Also we want to make server efficient enough to twig that there are no further players so fast in the next time and the game starts automatically.

By the possibility of adding pictures as question or to questions a lot of memory could pile on the server. So we want the pictures to be saved in a small format. This brings a lot more of memory and faster upload and download speeds. Also we want to avoid rollover questions or categories. We also thought that the public categories are hosted by us, but could make a the app a little be less interesting. So we want to make this a beta feature and the actions of the community will show, if we will keep this feature (to add categories as a user) or not. The up- and downvote features helps us managing the questions uploaded by the users.

The app itself won’t make many efficiency problems, because it’s held very low on CPU usage and doesn’t have any procedures that will slow the app.

Maintenance and portability

We want to release our app supporting German and Englisch, but in the future we might add some further languages to make the app more comfortable for different countries.

Our app brings actually a quite lot things we have to maintain. The probably confusing thing will be to maintain the community question and categories (if they will available to be created in public community), because everyone can add these. The more users the more questions will be there and the more confusing everything will get. So our idea was to introduce the voting systems where questions can be upvoted and downvoted and questions with a lot of downvotes and only a few upvotes will actually never come into a quizbattle round. Also it possible to report questions and when a questions will get reported (often) and there is really an issue with this question it will be removed, but this must mostly be checked by one of us (developers) or an community operator. Redundant questions or categories will be removed too.

Another point is that we always need to have a server running to make the game playable. We should have a backup server if one fails.

Additionally we want to get feedback of the users and improve the app and bring out some fresh updates when improvements are required.

Security

There is not very much trouble with data integrity, because the login information are all on Google, so this won’t make a problem. Also those question the user uploads, but that are not public to the community aren’t very sensitive data. So we actually don’t really have to care about. The only data, which might me a bit sensitive would be the email address of a user, but as there isn’t really any associated data it’s not tragic. We will try to make the server save and so that only the data from referring the requesting user can be received. The server can be accessed by operators (only us developers) through a login concept where a very long operator id and password are required. In addition there are only three tries per Ip address in a certain time interval to access the server. If too many login failures occur the Ip address will be blocked for a minute. Then are another three tries and then the blocked time will raise to a few minutes. Afterwards the ip will be completely blocked and reported and only an operator can set the ip free.

Legal

The only legal requirement we have is that we may only use pictures which are free to use or that are not copyrighted.

**4. Quantity Structure**

It’s a bit hard to really give a correct answer to the question “What number of expected records in master as well as business cases will come up”. This is because every question added by any user will be on the server. The same applies to the images uploaded by users and categories. So it can be seen, that the amount of data is dependent on the amount of users and on how much they add. Of course we must introduce a limit on how much data may be used by a single user, because we can’t offer endless memory.

We planned to use a database to handle all the data, because it will be very unstructured and unclear to simply use files. Three entities will be needed where the question are saved in. There are the “Guess”-Entity, “Category”-Entity, “Estimation”-Entity and the “Question”-Entity.

“Category”-Entity has the following columns:

* name (This is simply the name of the category)
* categoryNr (Every category gets a number e.g. first cat.:1, second cat.:2, …; will also be primary key)
* username (The user, that created the category)
* date (The date the category was added. For sorting purposes only)

“Question”-Entity has the following columns:

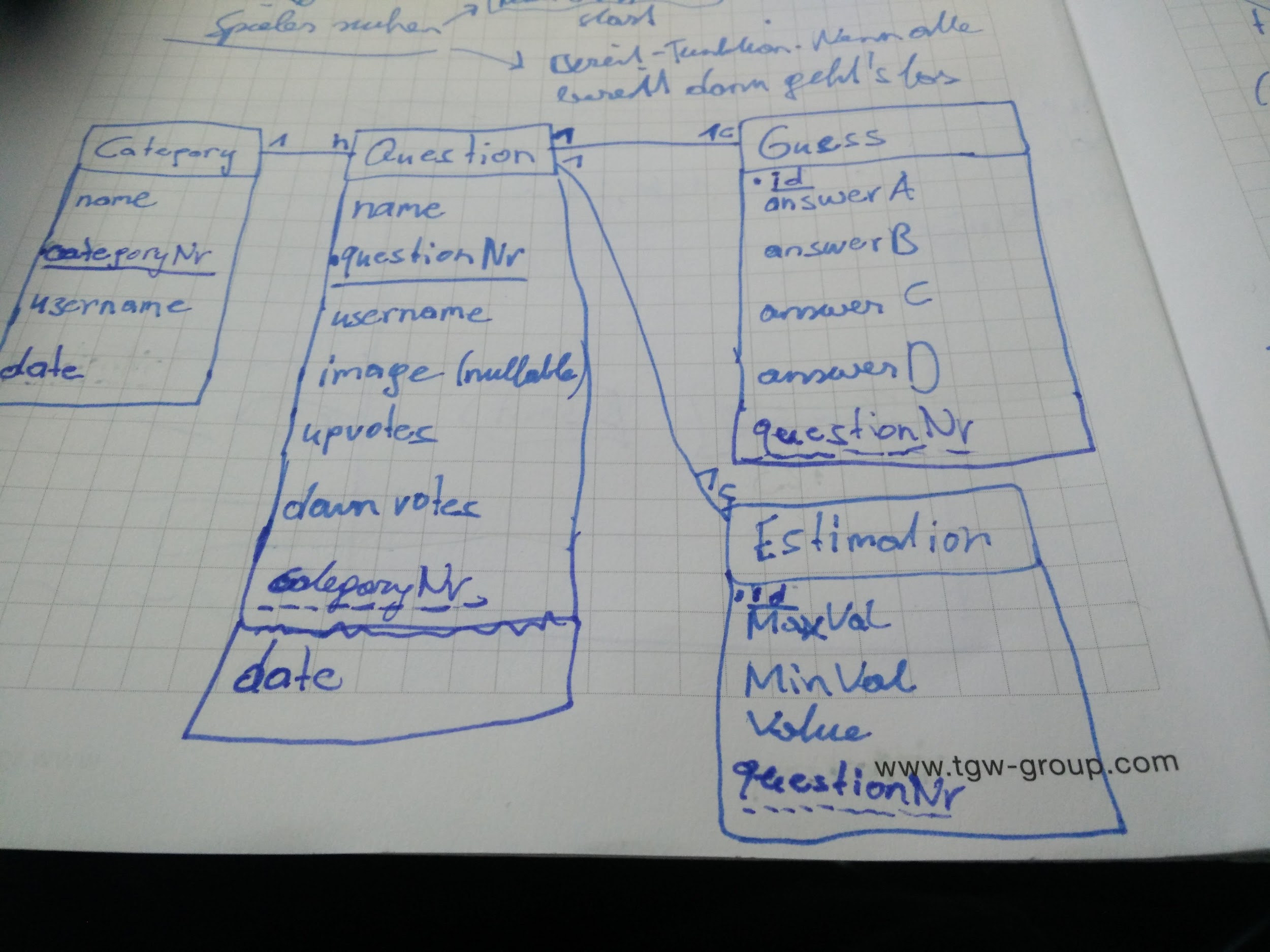
* name (The question text e.g. “Which state of the USA has the most inhabitants?”)
* questionNr(Every question gets a number e.g. first question.:1, second question.:2, …; will also be primary key)
* username (The user, that created the category)
* image (The optional image for the question; nullable)
* upvotes (The amount of upvotes the user gave this question)
* downvotes (The amount of downvotes the user gave this question)
* categoryNr (Foreign key to Category)
* date (The date the question was added. For sorting purposes only)

“Guess”-Entity has the following columns:

* id (Primary key)
* answerA (first option to answer the question)
* answerB (second option to answer the question)
* answerC (third option to answer the question)
* answerD (fourth option to answer the question)
* questionNr (Foreign key)

“Estimation”-Entity has the following columns:

* id (Primary key)
* maxVal (The maximum value the user can enter)
* minVal (The minimum value the use can enter)
* value (The exact value (answer) to the question)
* questionNr(Foreign key)

The following ERD gives a quick overview of the above explained entity columns.

Simple questions, answers and categories don’t need a lot of memory. The problem will be the pictures. We planned to cut big pictures down to a small certain size so that they don’t require a lot of memory.

We planned to have max room size of 40 players for one session of the game. So 40 players can play the same session at the same time. On the one hand 40 is a good number of max players because there might be for sure groups of players that could reach the size of 40. For instance a school class with 33 pupils could play. On the other hand there’s the server and the realtime multiplayer API (AppWarp). The API we are using seems to not really have a realistic maxium of players (100 players are still legit) which means that there could even be more than 40 players. So the only concern would really be that the server wouldn’t figure it out when 40 players are hitting a button at the same time but anyway that’s not really realistic because there can be a quiet big delay between the first player's button press and the last one’s. So the server can deal with such a big session and at this point we want to add that a full 40 player session will be rather rare. We also thought about the possibility that the host of a private session will also really host the session. What we want to say with that is that the host will take over the role that normally our server has and the host’s play device (smartphone, PC) will be the server of this private session. This would be something that would really lower the capacity utilization of the server but we are not 100% sure about this yet. We probably will introduce it, but keep it on a testing level and will not publish it, if it doesn’t really work out good.