# Feasibility Studies

For this feasibility study we’re going to bring up a couple of different approaches to the concept. We’ll also motivate methods that are implementable for our concept development both technically and from a designer point of view. Finally we’ll make a conclusion based on the discussed companies and/or methods that we’ve found during the research for this particular part.

As a little reminder to those of you who’ve forgot, our concept is based on distributing surveys and making them more fun and interactive, with the help of second screens. There’s going be both an inspirational/interesting visual representation based on the value of the survey. There’s also going to be an app that allows the user to both choose their answer to the posted survey, and a survey editor. As we’re going to market the concept as a fun way of interacting between each other through the public screen, we’ve chosen to add a lot of focus to the visual representations and the editor in the app.

Now that we’ve gotten reminded, let’s break the important parts down.

## Administrative approaches

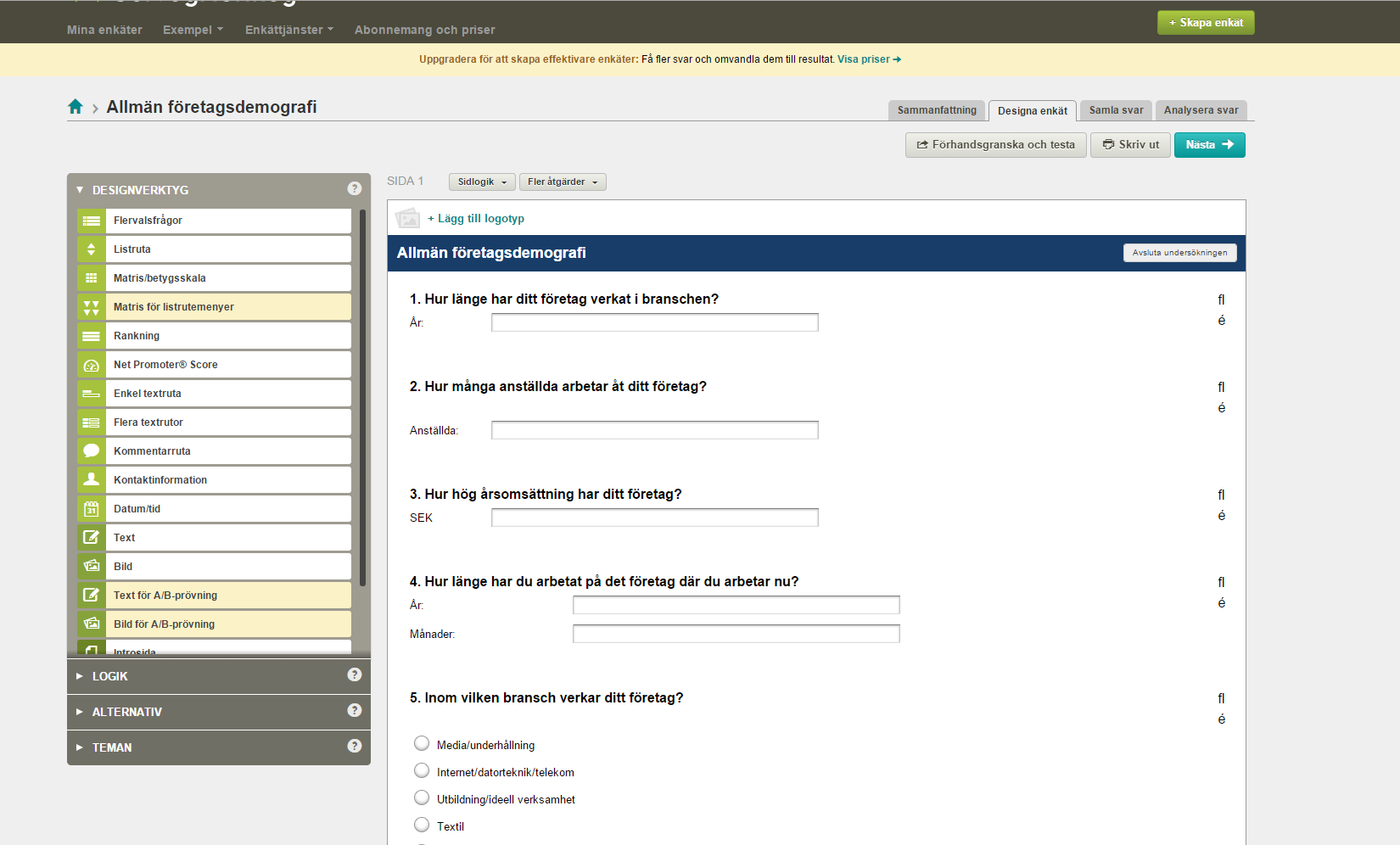
There’s multiple survey-apps and websites out there. We’ve taken a closer look at two of these: Straw Poll and Survey Monkey.

**Straw Poll** for example is a quick and easy way to throw up a survey with multiple choices. The way it works is simply that you fill in the question and a couple of answer and boom, you got a survey. You can then copy/paste an url to whoever you want to participate in the survey.

<http://strawpoll.me/>

This site shows us that the interface does not have to so extensive. As you can see in the picture above, there’s basically only 3 necessary interactive parts which are: Question, answers and the create poll button. This has given us good ideas on how to approach our app interface. We’ve taken this easy interface into account as we’re trying to take out too many options and just focus on the user being able to ask funny/interesting questions. Though the appearance might not be what we’re looking for, it shows us that it doesn’t have to be that hard to get a survey app up and running.

**Survey Monkey** on the other hand is a perfect example of providing many, many options in order to make your own personalized survey. Survey Monkey is a company that provides survey solutions easily accessible and to some degree, for free. However if you want to use expert templates you have to become a premium member.



<https://sv.surveymonkey.com/>

As you can see in the figure on top, survey monkey gives you plenty of choices and options to customize your survey to your own specific needs. Unlike Straw Poll that let’s throw out a quick survey, Survey Monkey caters to people who often need to do extensive surveys.

This survey creator gives us a great idea of what kind of options should be available in our product development, and also how to display it to the user.

## Technical methods and solutions

For the technical back-end solutions, easy put – back-end, we’ve decided to take a closer look at the app and swing-window provided to us by Lars Holmberg, teacher at Malmö Högskola.

What the app does, is that it tracks where you drag your finger across your mobile screen and then displays it on a larger window, which is a separate java application. The communication between the two programs works through something called Firebase. Firebase is a sort of cloud temporary storage that let’s connect applications or websites etc. over the internet. It makes for a simpler solution to network and storage problems. It also minimizes output lag that could very easily occur when you allow multiple users to connect to one single hub – in our concept, the big screen.

What we’ve looked at the most, is how the app connects through firebase with the java screen application. This is pretty much a must for our app to work properly. Also, since this code has already been provided to us, we can easily adapt the firebase connection as well as the screen to show us our graphical representation.

Another useful code example comes from two students at Malmö Högskola (Anton Nyman, Isak Lund Lilja) that provided us with an assignment where we we’re supposed to make an app that’s also connected to a secondary java application screen. In this, we had to make the app work as a “movie up voter” (in lack of a better explanation). In the app you got a movie cover with some additional info handed to you, and when you pressed the movie cover you voted on that particular movie. The vote then shows up in the java application window as a numeric vote, and when the movie you voted on gets popular enough, it changes the whole screen background to an image from that particular movie.

From this code we can easily pinpoint how to use a voting system together with firebase. Also what happens when a vote affects the screen and the visuals on that screen. We’re all quite familiar with the code as well, since this app and screen application was part of a school assignment in the course we are currently studying.

## Conclusion

Throughout the benchmarking process and this feasibility study we’ve come across a lot of helpful applications, code examples and design approaches to help us figure out the interface and back-end for our own concept. The question remains – is the concept feasible?

Based on everything we’ve been looking into above, the answer is definitely yes. We have a lot of the work already cut out for us. It’s all about putting bits and pieces together with the back-end. We have the competency within the group to make the back-end work. We also have skillful artists and a lot of creative ideas on how to work out the visual representations, and we’ve gotten great ideas on how to design the interface. Last but not least, we have the advantage of time at our disposal. Since a lot of the work lies in the code, and we’ve already got a lot of the “heavier” work done for us, we actually might have a lot of time to refine, reshape and extend the concept.