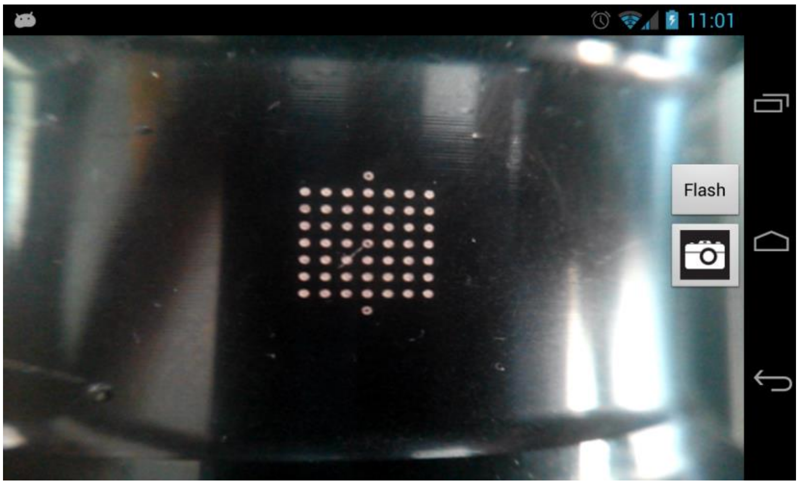
TensionCamApp: System specification

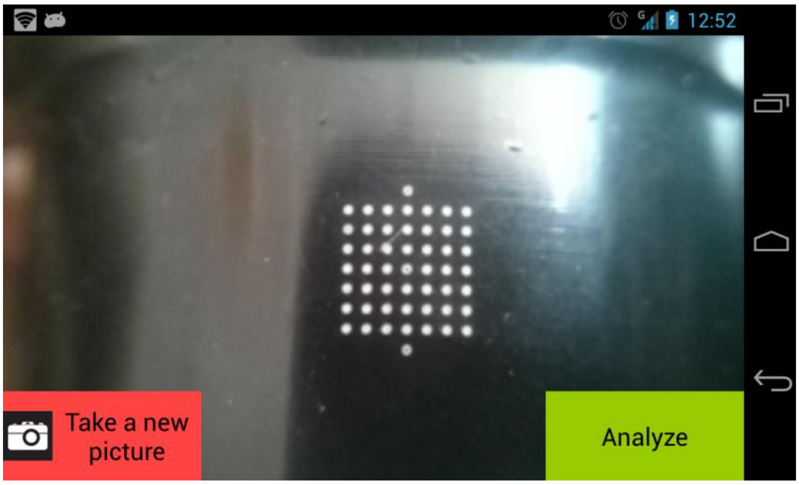
The app is designed based on our interpretation of our clients' requests. During the formation and development of the application we mainly had two areas of focus; easy to use with no room for errors and extendibility. Those two can be divided into two parts where the first one mostly relates to the layout while the second one mostly consists of logic and structure of the application.

Easy to use

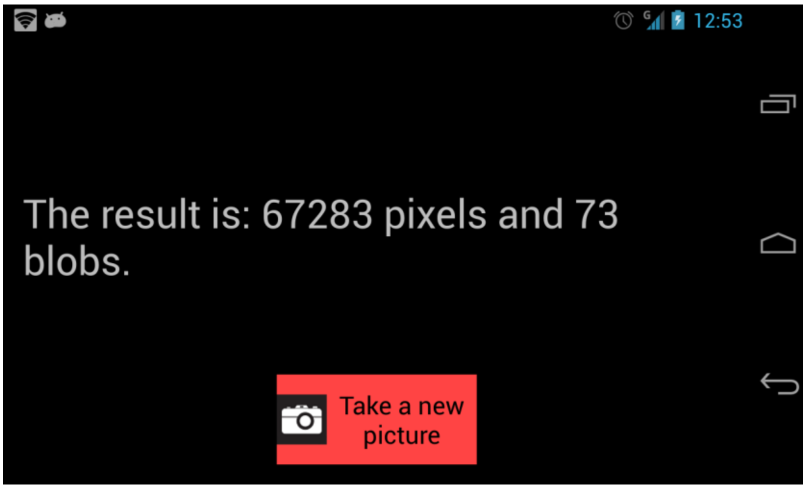
We aimed to minimize the clients' options and required inputs and only provide and present the functionality that every user needs with a workflow that is as simple as possible. This is motivated by that the user often works in demanding environments such as climbing in wind power stations and therefore don't have the time to mess around with the application. Another reason to minimize the users' choices is to insure the same output of the external analyse program for the same input. It shows foremost in the application by that we only included necessary views, which after revision turned out to be three, and the buttons required to go through the whole process.



The screen is locked into landscape mode to insure the same type of input every time. We also added the option to turn on the flash because it depends on the user other equipment if it's needed or not. We also tried to maximize the size of the picture to make it easier for the client to capture the desired picture and therefore to indirect give the best possible condition for the analysing program. That is why the buttons are on top of the picture. At the same time we kept the status bar in the view because it is important to see the internet connection since the process depend on it.



Our intent for this view was to give the user the best possible opportunity to review the image so we tried to make the layout look like the previous activity, camera mode. We figured that the users interest at this point is either to take a new picture if the recently taken one was unsatisfactory or to analyse it if the image were OK. There was a trade-off between the size of the buttons since we wanted to keep them small to increase the total size of the image but at the same time we wanted to make it easy for the user to find the right button, hence the choice colour. We also added the progress bar once the analyse button is pressed to show the user that the application is working, since with bad reception the analyse could possibly take a little while.



The result is: 67283 pixels and 73 blobs.

In the view showing the result we wanted a large text so that it would be easy to read. Because the application often is being used on dozens of bolts in a sequence we wanted to give the user the option to take a new picture whit out restarting the application.

Extendibility

The client asked for very specific functionality of the application, which wasn't at an extreme level of complexity. All though the client made it clear that the analysis project is one under development. Therefore we saw potential of extendibility in the future, which lead us to use logic and structure in the program which supported this. As a consequence some of the methods might seem redundant or unnecessary at the moment but is constructed with the motive to be able to develop further.

One of the biggest design decisions that were influenced partly by the extendibility was to put the analysing programme at an external platform. Not only is it easier to maintain and update but it also gives the option to add new features to the web server without requiring the user to update the application on the local device. All in all it decreases the coupling in the whole system and at the same time increases the cohesion since the application on the device integrates with the user and the web server handles the logic.

An aspect we took under consideration while developing the application was future needs. In the case of commercialization of the app we predict that there might be an interest in storing data externally on a part that is accessible for all users. This so that previous data for a bolt may be accessed by different users. Because of the architecture currently in use and the fact that an external web server is used this is an expansion is easily carried through if desired.

Use of web server

Above a number of advantages are presented which comes with using a web server. Further positive effects that contributed to the decision are mainly: Less redundancy and better process power.
Less redundancy is achieved by the fact that every local device (cell phone) do not need to carry the analyze program. This minimizes the storage need for the application. The development of faster telephones is a fact but are nowhere close to be compared whit a computer. By using an external web server it is possible to perform advanced and complex analysis considerably faster than a phone would be able to.

The web server also enables future additions. For example it is possible to store all results from different users on the server and perform comparisons and statistical evaluation.

Architecture

Throughout the project we have strived to keep the dependencies between different classes and packages low. In order to verify low dependency, a structure analyze of the android application was performed. Figure 1 illustrates the result of conducted structure analyze showing that utils., views. and controllers are not dependent on the activities or each other and are not affected by changes made in these packages.

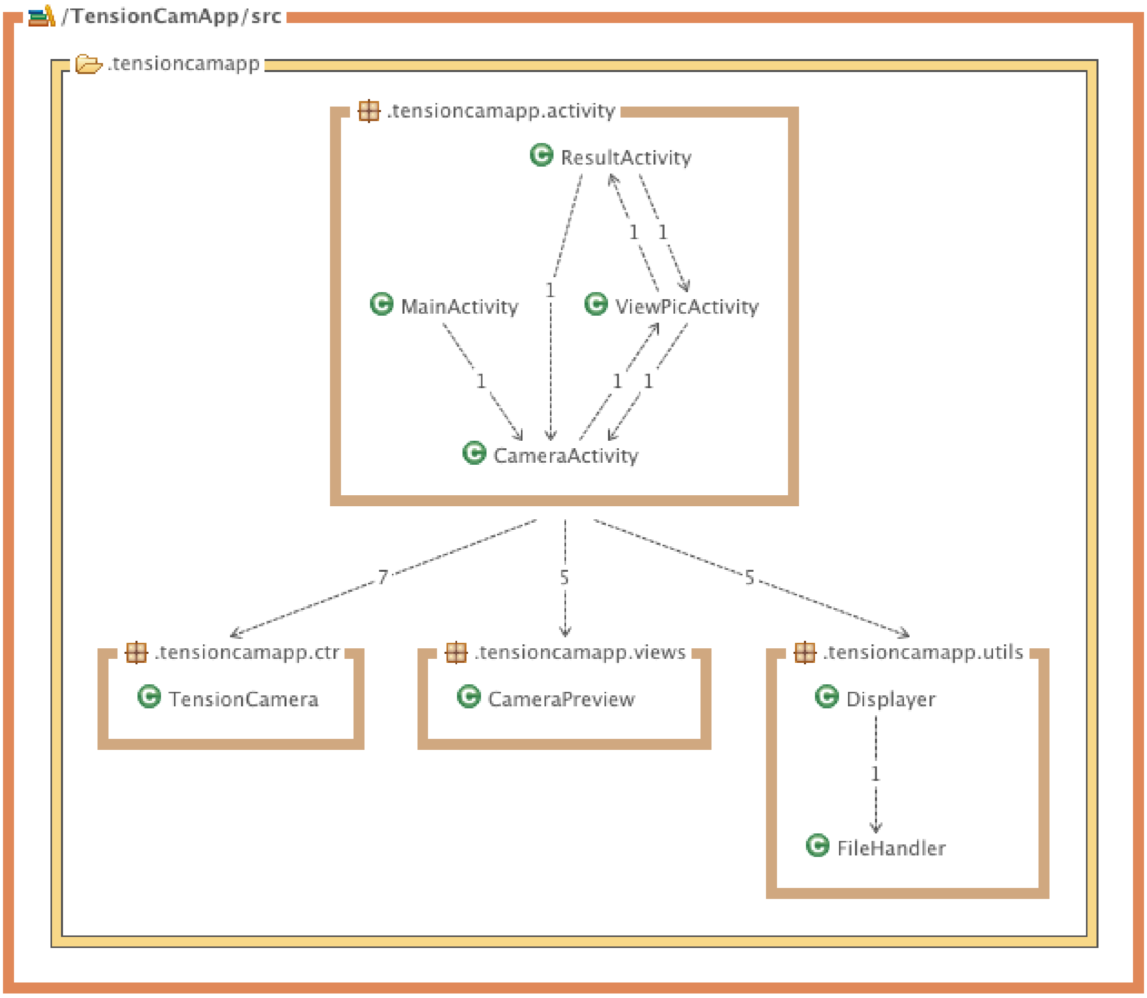


Figure 1. Structure Analyze

On the web server the architecture is based on the Spring MVC framework. That is an architecture based on controllers and views. This enables an easy use of, in our case, http request. Additional classes are added to support the controller class. This enables construction of as slim and effective controller classes as possible. It also reduces the complexity of the controller methods.