
Title goes here

Developing complex software systems

Aalborg University
Software – 2015
Project group SW605F15



AALBORG UNIVERSITY
STUDENT REPORT

Department of Computer Science
Aalborg University
Selma Lagerlöfs Vej 300
9210 Aalborg East
<http://cs.aau.dk/>

Title:

Title name here

Theme:

Developing complex software systems

Synopsis:

Project:

6th semester project

Period:

February - May 2015

Group:

SW605E15

Authors:

Claus Nygaard Madsen
Lukas Nic Dalgaard
Martin Juul Johansen
Sean Skov Them

Supervisor:

Bin Yang

Copies:

Pages:

Appendix:

Finished: 29-05-2014

The content of the report is freely available, but publication (with source reference) may only take place in agreement with the authors.

Signatures

Claus Nygaard Madsen

Lukas Nic Dalgaard

Martin Juul Johansen

Sean Skov Them

Preface

Reading guide

Contents

1	Introduction	1
1.1	Status of GIRAF	1
1.2	Multiproject, Subprojects, and Responsibility	1
1.2.1	Database	1
1.2.2	Graphical User Interface	1
1.2.3	Build and Deployment	2
1.3	Project organization	2
2	Sprint #1	3
2.1	Google Play and Analytics	3
2.1.1	Google Play	3
2.1.2	Google Analytics	3
2.2	Javadocs	4
2.3	User Stories and Tasks	4
3	Sprint #2	7
4	Sprint #3	9
5	Sprint #4	11
6	Collaboration	13
7	Conclusion	15
	Bibliography	17
A	Google Analytics Crash Reports to Email Guide	19

Introduction

1

GIRAF is a series of applications(apps) for Android, intended to help citizens with autism in their everyday life. Over the past four years GIRAF have been developed by students at Aalborg University, with a new group of students taking the mantle every year. As a result of this, it was hard to get an overview of the project in its entirety. We here give an overview of the GIRAF project when we started working on it, and we also give some insight into the organization of the project.

1.1 Status of GIRAF

Overall GIRAF is in a somewhat functional state with several problems ranging from small to severe regarding stability, reliability, and usability. GIRAF is published in Google's Play Store with the individual apps being version 1.0 while the launcher-app is in version 2.2. The version-numbering is not entirely correct, as some of the apps does not function correctly yet and as such they should not have been published as official versions but rather as alpha or beta versions. The majority of the apps does function correctly, but only a few of them are ready for public release. Google Analytics was incorporated in roughly a third of the apps, with between three and six months of data on usage and crashes being available. The data has been gathered since the newest version of the apps were published, in general since the 25th of August 2014.

1.2 Multiproject, Subprojects, and Responsibility

The overall mission for the multiproject this year was, to make a working system that is usable and stable. To accomplish this, the project was split into three subprojects: Database (DB), Graphical User Interface(GUI), and Build and Deployment(B&D). We were a part of B&D.

1.2.1 Database

DB groups were responsible for handling all things database related, this included making current data accessible and adding new data to the database, as per request from other groups or as needed.

1.2.2 Graphical User Interface

GUI groups were mainly responsible for developing the different apps, this included making general changes in the apps and bug fixing anything related to the apps.

1.2.3 Build and Deployment

B&D groups were responsible for most of the internal project matters, this included maintaining the server services, developing and maintaining the automatic build tools, handling publication of the apps, and administrating the version control.

1.3 Project organization

The project was organized using Scrum principles, where the individual subprojects held Scrum meetings twice a week. The individual groups could follow any principles that they would like, in our group we followed Scrum. In addition an overall meeting was held every week where all groups were present. At these meetings the overall status for each subproject were given, and any decisions that needed overall consensus were discussed.

As the overall structure was Scrum, the project was split into four sprints. The first sprint was devoted to bug fixing, user stories and tasks that were still unsolved from last year, and setting up and organizing project tools. In regards to user stories, it was decided early on at an overall meeting, to operate with two types of user stories, those from the costumers, these were called user stories; and those from the other groups, called developer stories. For consistencies sake we will just refer to both as user stories in the report, as the difference was mainly a concern when groups from different subprojects where discussing internally.

Sprint #1 2

For the 1st sprint our main user story was setting up and continuously support Google Play and Google Analytics. We also had a user story to write guidelines for Javadocs, and we were given the role as Product Owners for B&D.

2.1 Google Play and Analytics

This section describes our work with Google Play and Google Analytics.

In order to setup and support Google Play and Google Analytics, we were given the administrator login to both services. In the following we describe our work with the two services.

2.1.1 Google Play

Google Play is a digital distribution platform created by Google. It is used as an appstore for android devices as well as for music and e-book distribution. Because we in the project are making an apps for android devices, we are using the Google Play store to distribute the different apps.

As it was our responsibility to maintain and control Google Play for this semester, we started by familiarize us with the different features that Google Play provides. We found that Google Play provides a lot of statistics for all the apps, like how many times the app has been installed and how many are still using it. Like having access to statistics, Google Play also allowed os to change with apps should be shown on the store and to change the descriptions for the apps. One thing we noticed when we looked at some of the description was that a lot of the descriptions was very short and did not provide enough information about the apps for the user to know what functionality the apps provided. Because of this we decided to update some of the descriptions, make an English version, and to update some of the pictures provided for the user.

One data that Google Play provided with was very important was the statistic about with version of android the users where using. because from that we found that the apps has to use API 15, because 20% of the users where using android version 4.0.3 - 4.0.4.

Beta

2.1.2 Google Analytics

Google Analytics is a service created by Google. The service generate statistics from traffic on web. Google Analytics main function is to track visitors and display advertising, to show

how people are using the products. Mobile App Analytics is a branch of Google Analytics, which focus on mobile. Mobile App Analytics has google play integration which allow the visitors/traffic sources tracking. this traffic can then be use to see who is installing the apps on which platform. Mobile App Analytics also provide some API's for tracking events, Crash, Exception, and user patterns, to see how the users use the app.//

Mobile App Analytics and Google play have a lot of the same functionality for tracking who installed what on which platform. But Mobile App Analytics has an advantages when it comes to crash/exception rapports, because API's allow for reporting without the user has to interact and provide more detailed information on where the crash/exception happened in the code, therefore Mobile App Analytics were used to report crash/exceptions.//

For crash reporting there were three task at hand. the first was to autoforward the crash reports for a submodule to the groups that worked on that submodule. The second and third task was to provide a guide on how to implement the API's into the apps and provide a guide for the next years Google Analytics group on how to auto forward crash reports. //

Autoforward crash reports

Mobile App Analytics has an build in Email feature, which allow the user to auto email a crash report daily, weekly, monthly or quarterly. It is not that intuitive how to auto forward reports because the user has to first make an report that a able to filter to the time interval the user wants. Therefore an guide was made which can be found in the appendix.

Mobile App Analytics API

The Crash API for Mobile App Analytics has to be implemented if crashes are to be fund. The API works by making Tracker which can be used to report information to Mobile App Analytics, when an exception is caught. An great feature about the tracker is that it can be set to report uncaught exception.

2.2 Javadocs

2.3 User Stories and Tasks

Z

Sprint #2 3

Sprint #3 4

Sprint #4 5

Collaboration 6

Conclusion 7

Bibliography

Google Analytics Crash Reports to Email Guide



This is a step for step guide for the setting up crash reports from Google analytics to email, for the google analytics group of 2016.

1. The first thing to do is to go to the google analytics website at <http://www.google.com/analytics/>
2. Then log in to the google account using the google account information provided.
3. The page should then look like shown below on figure A.1.

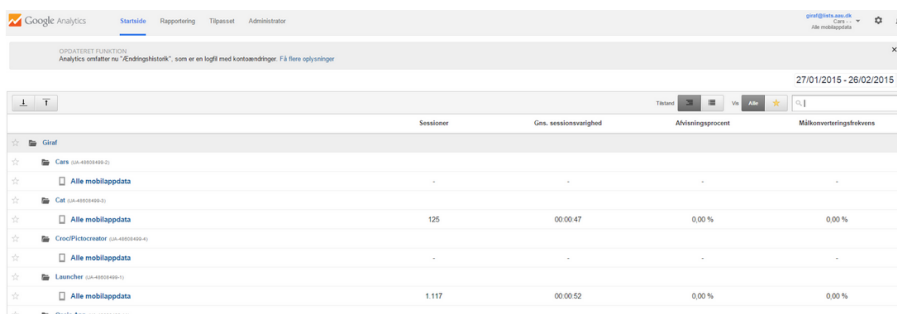


Figure A.1. Main page when logged in to Google analytics

4. Select the app you want to have a crash report from by pressing “Alle mobile appdata” Right below the app.
5. This should then lead you to a page like on figure A.2 the one below.

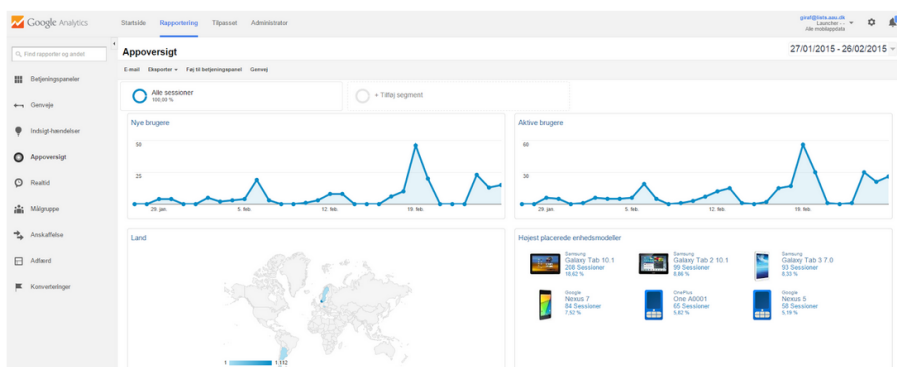


Figure A.2. App overview page

6. At the left most sidebar, press on “Adfærd”/”Behavior” and select “Nedbrud og Undtagelser”/”Crashes and Exceptions” in the dropdown menu.

7. The Page should then look like shown below on figure A.3.

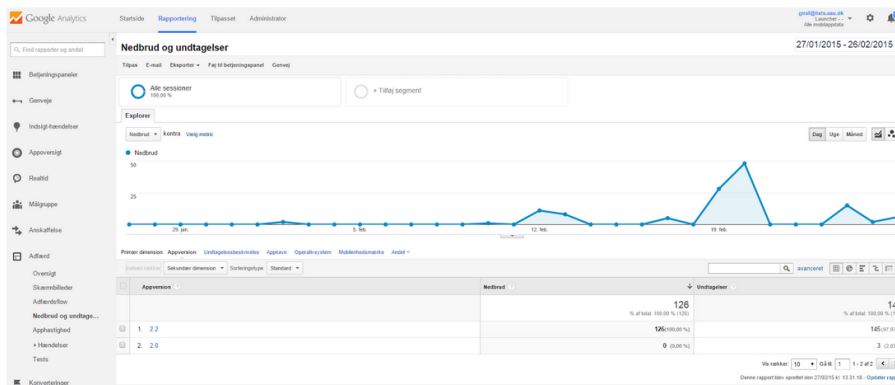


Figure A.3. Crashes and Exceptions page

8. At the bottom of the page click on the newest version of the app.
9. The page should then show all the crashes at the bottom of the screen.
10. After this go to the top right of the page and click on the date interval.
11. This should open a window like the one shown below on figure A.4.

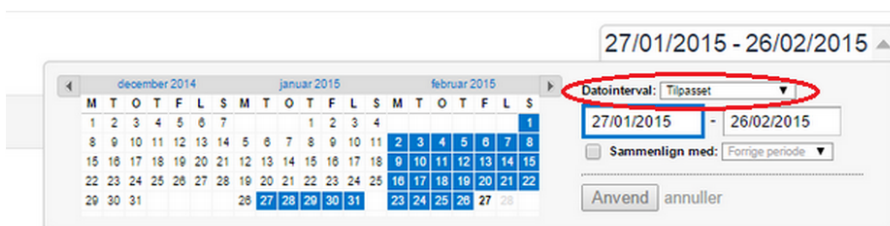


Figure A.4. Calendar window

12. In the window select the time interval that you want the crash report to be sent from in the highlighted dropdown menu.
13. Press “Anvend”/”Apply” to close the window.
14. In one of the top menus there is a button named “Email”, press it.

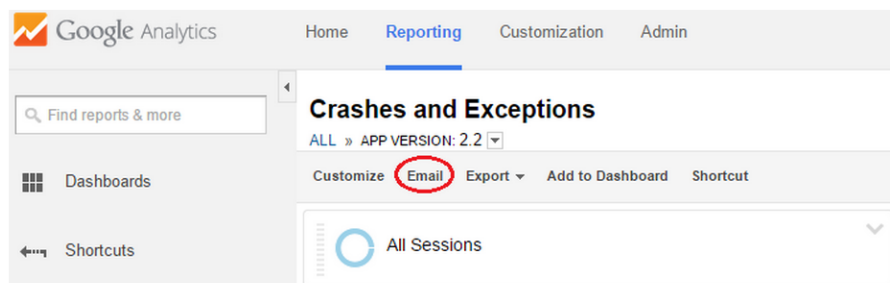


Figure A.5. Email button

15. In the window that opens select whom to send the mail to, what format the crash report should be in, and when the email should be send.

E-mail-rapport: Nedbrud og undtagelser

Fra giraf@lists.aau.dk

Til

Emne Google Analytics: Nedbrud og undtagelser

Vedhæftede filer CSV NEDBRUD OG UNDTAGELSER

Frekvens Ugentlig Ugedag: SØN MAN TIR ONS TOR FRE LØR

► AVANCEREDE INDSTILLINGER

Send Annuller Føj til en eksisterende e-mail

Figure A.6. Mail report window