



HoGent

Faculteit Bedrijf en Organisatie

Research of caching strategies in mobile native applications using external data services

Bijlagen

Frederik De Smedt

Scriptie voorgedragen tot het bekomen van de graad van
Bachelor in de toegepaste informatica

Promotor:
Joeri Van Herreweghe
Co-promotor:
Jens Buysse

Instelling: —

Academiejaar: 2015-2016

Tweede examenperiode

Faculteit Bedrijf en Organisatie

Research of caching strategies in mobile native applications using external data services

Bijlagen

Frederik De Smedt

Scriptie voorgedragen tot het bekomen van de graad van
Bachelor in de toegepaste informatica

Promotor:
Joeri Van Herreweghe
Co-promotor:
Jens Buysse

Instelling: —

Academiejaar: 2015-2016

Tweede examenperiode

Contents

1	Introduction	3
1.1	Problem definition	3
1.2	Research questions	4

Chapter 1

Introduction

The definition of mobile has changed a lot in the last few years and the expectations of mobile applications continue to rise. Mobile applications should work on a myriad of devices and screen resolutions, and should take battery efficiency into account. Furthermore is an application no longer really considered mobile if it isn't available without a stable internet connection. This makes research about caching in mobile applications really interesting since it can solve multiple problems. Caching of the results of expensive operations, e.g. an internet connection, will allow the system to avoid many of these operations, reducing battery consumption, and can these results be used when a (temporary) connection loss would occur.

Lots of IT companies already implement such a system:

- The Facebook-app still shows several posts and pictures when you have no internet connection;
- Third-party API's, such as Picasso (Android) and SDWebImage (iOS) retrieve images from the web and have an embedded caching system;
- Twitter allows you to change account settings offline, which are later synchronized with the backend when a connection is established¹.

1.1 Problem definition

However mobile caching is becoming more and more common, there is no real framework on which they all rely. Instead they each have to think about how to implement

¹In this example we are talking more about persistent data, however it is used to store temporary data that can later be forgotten once it is synchronized with the backend. Just like a cache containing business data.

caching and have to invest in discovering caching strategies and inventing a good caching implementation.

They can decide to ignore the caching problem and focus on the business logic and the actual functional requirements of the project. The chances that the project is able to be completed within the deadline increases, as they have less work to do. However, if there is a non-functional requirement that requires some sort of caching or if the users of the application are complaining about problems that come along (e.g. lots of traffic or battery usage), you will have to implement the caching system afterwards. Which might force them to redesign parts of the system (both frontend and backend) or forces you to do this in an incomplete way, introducing lots of bugs that usually come along in caching systems.

1.2 Research questions

Every platform already has some infrastructure designed to allow caching, e.g. Guava and a local lightweight database. Yet there is more to caching than simply storing information, developers should think about what data is eligible for caching and how this can improve their application. They should be able to do this with an efficient method, customized to the needs of the application considering all possible events that might occur and how they might be able to handle this. These ideas and strategies should then be implemented in both the backend and frontend, however the backend implementation could be reduced to an almost non-existing implementation, since it is not merely about the native application itself, but about the data flow between the different systems.

Because of these reasons, we will try to answer the following questions in this article:

- How can a caching strategy efficiently store and fetch data in a native mobile application?
- What life-cycle events should be considered and how can they react to these events?
- How can the cache be synchronized with the external data service (backend)?