Bachelor Degree Project

(Working title)

Automatic synchronization and data merging between iOS and backend

- Solution for easy setup of synchronized CRUD functionality between backend and application

Author: *Mikael Melander*
Semester: ST 2018

Abstract

Keywords: iOS, crud, Linux, offline synchronization

Preface

Contents

1 Introduction (draft)

This degree project will study and develop a solution for offline data handling and synchronisation with and online database.

The study aims to examine the best rules for data merge solutions as well as data structures that can be handled within both iOS and an online database solution.

1.1 Background

The world of mobile applications has exploded in the last couple of years. Companies increasingly adopt the functionality of mobile devices to streamline the daily workflow, what used to be done by pen and paper or only by sitting down at a computer is now possible with mobile applications.

Because a mobile phone can be taken anywhere, your work should be able to follow. But most applications demand an external database to store data across users, this intern relies upon a mobile data connection or WIFI by the device itself.

This can become a problem if you find yourself in places where your cellular reception is limited. If this happens then you won't be able to work. The solution to this is to be able to add, see and use the data that already exists within the application regardless of your cellular reception.

1.2 Related work

Research around similar solutions have been conducted and there are some big and famous companies that have created some type solutions to the problem. These are some solutions that works really well and integrates the solution that this project aims to solve. They have some back draws and exactly how they solve it isn't disclosed.

Firebase is Googles database service that allows you to create an mobile/web application connected to the database within a couple of steps. They have a complete framework for iOS, android, windows phone and web solutions. They also have a set of tools to verify data, create security, analyse usage, send push notifications and many more. But Firebase as a Google solution locks you down to their own way of thinking, you need to use it on their premises or not at all. You can only store data on their database and build your solution around them.[1]

Microsoft Azure also supports the ability for offline client sync, but as well as Firebase it relies on you using their cloud services[2].

This subject is a mace and there are a lot if different solutions that none of the are exactly the same, the have different end goals and work with different already existing server platforms.

I also found some solutions that work for frameworks such as Xamarin that is a cross platform .NET solution, react-native that is a cross platform JavaScript solution and one that is for ionic that is a JavaScript cross platform solution that works with html5 instead of native as react-native does[3][4]. These solutions work with specific server platforms, and some of them only locally duplicating the database that is online, not handling the merging situations that the server needs to be able to handle. Most solutions that I found aim towards integrating an offline synchronisation solution for an already existing platform, as a layer to support the platform. What this project aims to do is to integrate the offline solution as a bottom layer and build the platform upon that solution instead of the other way around.

1.3 Problem formulation

The goal is to find a suitable solution that can handle objects structures on both the iOS locally and the backend database the same way.
Creating the offline functionality by saving cached data on the device and the synchronising it to the backend when the connections allows it.

Because the solution should be angled towards multi tenant usage, the system itself needs to be able to determine the correct way to merge different versions of the same object trying to be saved to the database[3].
In order for the system to be able to track this, the data model needs to be implemented in such a way as to allow this. For example, adding version values to the object.

1.4 Motivation

Today it already exists different solutions and platforms for offline data synchronisations, for example Firebase.

Firebase is owned by google and using it locks you down to their backend and systems and you have little to no control over your data.
Many companies today that want a solution for mobile applications associated to their work flow would not allow that the data is stored anywhere other that in their own control.
So the creation of a ready to go backend solution that runs on a Linux environment and can be easily connected to an iOS device would create an opportunity for these companies to be in control of the complete backend.

1.5 Objectives

O1	Research and determine server platform
O2	Create server database and iOS database model
O3	Implement connection between iOS and server
O4	Implement local storage iOS that handles the database model
O5	Implement methods for querying data given specific arguments that handles local/server storage
O6	Implement functionality that automatically tracks what data has been sent to the server and what data only exists locally
O7	Determine and perform experiments for merge rules for data

Because of the need to be able to implement querying and rules for merging of data, the database needs to be run behind a server.
The intent is to research and determine a suitable and reliable server framework as well as a database model that will suit both the local storage and the server database.

For the offline data to work the device need a local storage, the storage should work as a middle layer to the server database query. When new server data is returned it should be saved locally, then returned. The implementation needs to keep track of what data is already stored locally and what data is new and needs to be saved.

The server itself needs to be able to handle the requests and execute merge rules.

1.6 Scope/Limitation

Within the scope of this project, the solution should allow for a relatively easy setup that gives you a reusable solution within iOS/Xcode that would allow you to query data and return results, either from locally saved data or from the online database depending on your internet connection.

You should be able to save new data and edit previously added data that when reconnected to a internet connection should be synced with the database.

The solution that is created and the dependencies used for it has to be free.
Because Xcode and iOS changes over time the solution should be set to work and support at least the latest 3 versions of iOS (9-11).

This project will not take in to consideration the security aspekts meaning that it will not have e ready solution for HTTPS or that it will have any users or data access lists.

The project will not support real time database.

1.7 Target group

This project can be of interest to companies, organization or persons wanting to host their own solution that integrates with iOS that is cost efficient.
The solution could as well be used as a starting point to keep building upon, but already has the important support for offline synchronization.

1.8 Outline

The next chapter will present the **methods** that is used to execute the different objectives that is presented above.

The **implementation** chapter will be a more detailed explanation of how the project will be implemented and how the solution itself works, how the merge rules will be executed and how the automatic syncing is handled.

The **result** chapter will cover what came of the project, what the resulting structure of the solution became.

The **analysis** will cover an overall analyze of the concluded results.

The **discussion** will deeper discuss the analysis and results.

Chapter seven will include **conclusions** that are based on the results as well as present future work and recommendations.

2 Method (draft)

The method used to conduct this project will be verification and validation.

The project is not created in any collaboration with a company, meaning there is not a given outline from an external source to create these requirements. So to get the requirements for this method, the defined problems for the project will be converted into functional and non-functional requirements.

This project does not build upon already existing code or will not use any existing code that will have to be collected (This does not include the frameworks and platforms that will have to be used). This means that the functionality of the project will be based upon written code for the functions that needs to be implemented, so the requirements will make sure that the implementation and functionality is correct.

By using the verification and validation method we can see if the project has been correctly created as well as is a successful valid solution.

2.3 Reliability and Validity

To be able to use the verification and validation method correctly and be sure that the results is reliable, the requirements we create needs to be measurable and objective. This means that we need to make sure that the requirements can't be subjective, if they are subjective different people can interpret the requirements in different ways. If this would happen the reliability of the results could be compromised.

The requirements created from the problem definitions will broken down in small pieces that will be easier to understand and phrased in a way that should be easy to confirm or deny if it is fulfilled or not. Conducting it in this way will help to ensure the reliability that the verification and validity is correct.

The verification and validation method is most often used when you want to confirm the results of a working project to a customer. But because I work alone the validity of the results might be questioned. This project aims to create a solution that I myself want to exist, to use and work with. There would be no reason for the results of the verification and validity to not be correctly conducted.

2.4 Ethical Considerations

The project goal is to create an easy deployable server, iOS and offline data synchronization solution. The solution should be open sourced and has a potential to be worked on more to create extra functionality and widen the scope.

By conducting this project I can see no reason for any ethical issues to come to light.

3 Implementation

4 Results

5 Analysis

6 Discussion

7 Conclusion

7.1 Future work

References

[1] https://firebase.google.com/docs/database/ios/offline-capabilities
2018-02-29

[2] https://docs.microsoft.com/en-us/azure/app-service-mobile/app-service-mobile-ios-get-started-offline-data
2018-02-29

[3] http://opensourceforu.com/2017/01/mobile-app-without-internet/
2018-03-01

[4] https://github.com/sqlite-sync/SQLite-sync.com
2018-03-18

[5] https://frontmag.no/artikler/utvikling/offline-data-synchronization-ionic
2018-03-18