

Smart Life

COM S 402 SENIOR DESIGN DOCUMENT

Team Number: sd_3

Client: Simanta Mitra
Advisers: Simanta Mitra

Team Members/Roles
Dengyun Ma – Frontend Design
Wentao Pei – Frontend Design
Qinwen Yang - DS Image Recognition
Xin Wang – Backend Design

Team Website:
<https://git.linux.iastate.edu/seniordesigncoms/sd3/wikis/home>

Revised: 5/3/2020

EXECUTIVE SUMMARY

Development Standards & Practices Used

- GitLab
- CI/CD
- Mockito test
- Android Studio

Summary of Requirements

- Allow users to set alarm and choose different types of wake-up method to ensure the user really get up from bed.
- Users can set up new event on the calendar.
- Observer users could setup events or alarms for normal users.
- Observer users could receive message when normal users turn off the alarms that he/she set.

Applicable Courses from Iowa State University Curriculum

- DS201 – Introduction to Data Science
- DS301 – Applied Data Modeling and Predictive Analysis
- ComS309 – Software Production
- ComS227 – Introduction to Object Oriented Programming
- ComS228 - Introduction to Data Structures
- ComS362 – Object Oriented Analysis and Design
- ComS363 - Introduction to Database Management Systems

New Skills/Knowledge acquired that was not taught in courses

- Create CNN Model for Image Processing
- Connect the Front-End with data

Table of Contents

1	INTRODUCTION	3
1.1	ACKNOWLEDGEMENT	3
1.2	PROBLEM AND PROJECT STATEMENT	3
1.3	OPERATIONAL ENVIRONMENT.....	3
1.4	REQUIREMENTS.....	4
1.5	DESIGN ASSUMPTIONS AND LIMITATIONS	4
2	ARCHITECTURAL DIAGRAM.....	5
3	COMPONENTS.....	6
3.1	COMPONENT-1	6
3.1.1	<i>Module 1</i>	6
3.1.2	<i>Module 2</i>	7
3.2	COMPONENT -2	7
4	DATA DECOMPOSITION	8
5	DETAILED DESIGN	9
6	DESIGN RATIONALE	10
6.1	DESIGN ISSUES.....	10
6.2	< ONLINE ALARM SAVING FAILED>	10
6.2.1	<i>Description</i>	10
6.2.2	<i>Factors affecting Issue</i>	10
6.2.3	<i>Alternatives and their pros and cons</i>	10
6.2.4	<i>Resolution of Issue</i>	11
6.3	< IMPLEMENT CALENDAR VIEW AND ALERT DIALOG IN FRAGMENT >.....	11
6.3.1	<i>Description</i>	11
6.3.2	<i>Factors affecting Issue</i>	11
6.3.3	11
6.3.4	<i>Alternatives and their pros and cons</i>	11
6.3.5	12
6.3.6	<i>Resolution of Issue</i>	12
6.3.7	12

1 Introduction <NOT TO EXCEED TWO PAGES>

1.1 ACKNOWLEDGEMENT

We would like to thank our professor, Simanta Mitra, act as our mentors, help us finding external sources for our project, provide us valuable advice and feedbacks.

We would also like to thank Dr. Thomas Swartwood and Adisak Sukul. Dr. Thomas Swartwood has offered us lectures about entrepreneurship. Adisak Sukul has provide advice and help for our DS student.

Finally, we thank all the members of sd3 for their efforts.

1.2 PROBLEM AND PROJECT STATEMENT

SmartLife is an app that helps everyone improve their daily schedule. In this app, we will have an alarm setting and calendar setting that allows all the observers and normal users to set the alarm or edit the calendar schedule. Observers and normal users are able to set an alarm or edit calendar schedule to themselves. In addition, observers are also able to set the alarm or edit calendar to their users, arrange their tasks. Our team discovered that forcing people to do some activities in the morning could effectively help people to wake up. Therefore, our app provides several special ways to help people wake up in the morning. For example, we implemented a shaking activity for the alarm. Once your alarm ring, the only way to shut down the alarm to shake your phone, which sounds a little bit annoying, but it is actually helpful for waking up yourself.

Unlike most alarm application on the market. Our team is focusing on providing online service for our users. To be exact, we design our application to have two kinds of users, observer, and normal users. If you are a parent, you could use this app to schedule events and time for your children.

1.3 OPERATIONAL ENVIRONMENT

The application is designed for android mobile devices. It can either offline use or online use. User can set alarm or calendar to themselves as offline use. For online use, user can set alarm or calendar to themselves, but also set alarm or calendar to others.

1.4 REQUIREMENTS

Functional requirements:

- Alarms
- Different types of waking alarms
- Setting up calendar events
- Online Functions (Register, Login, Building relationships, etc.)

Non-functional requirements:

- Availability
- Reliability
- Recoverability
- Maintainability

1.5 DESIGN ASSUMPTIONS AND LIMITATIONS

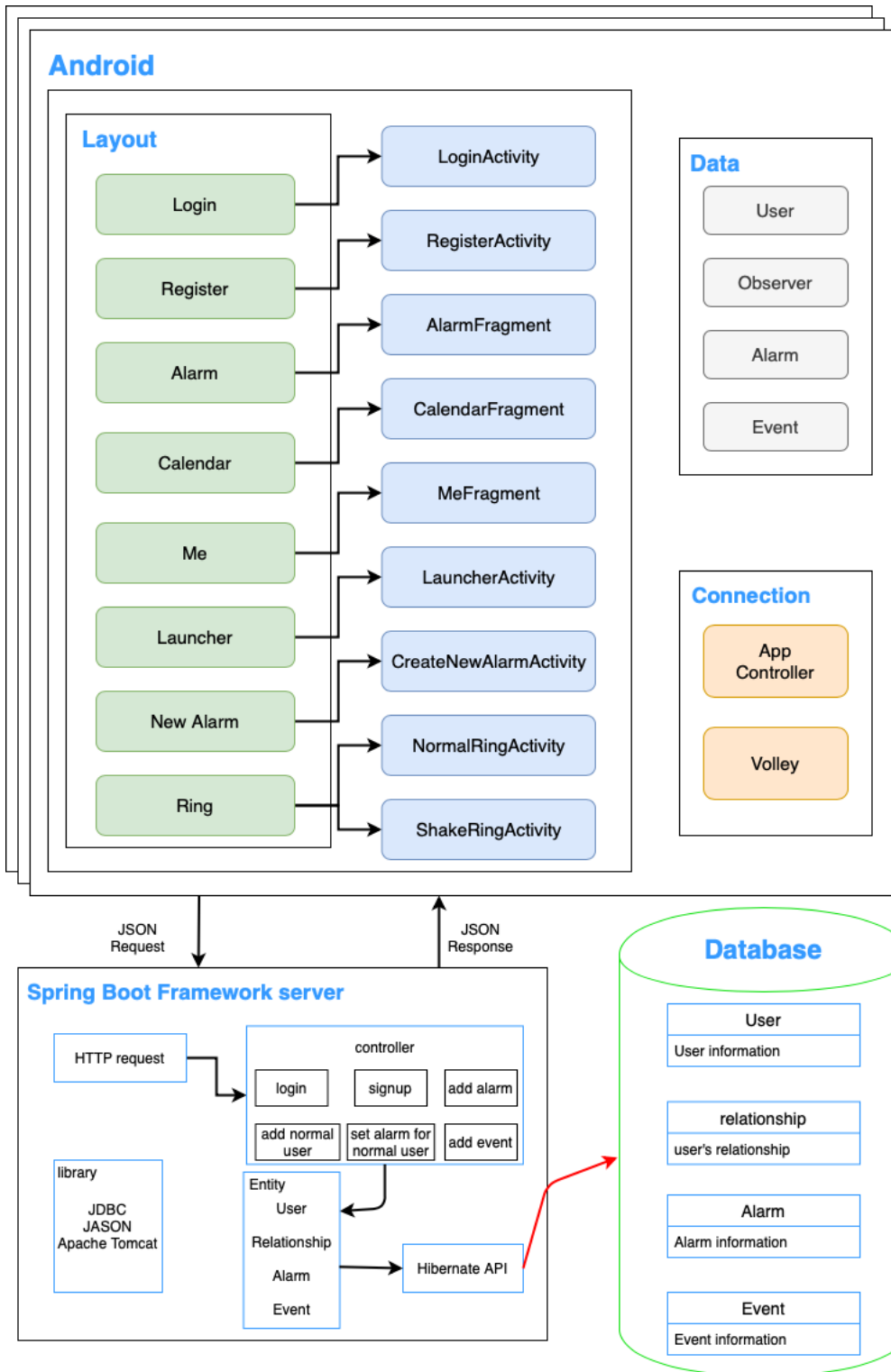
Assumptions:

- The application is available on Android platform as a portable alarm clock.

Limitations:

- Calendar events could only set up in exact date but not exact time.
- Alarm clock could only set for next 24 hours.

2 Architectural Diagram



3 Components

This application has two major components, the first one is the client on Android platform, the second one is the backend server that built with spring boot and MySQL.

3.1 COMPONENT-1 - CLIENT COMPONENT

The Client Component has four major modules. Alarms setting module, calendar module, personal setting module, and the alarms activity module. The first three module are set up separately in three different fragments of the application. The last one will trigger only when it is the time for alarm to ring.

3.1.1 Module 1- Alarms setting module

The alarms setting module allows users to set up alarms and choose their favor way of alarm activity to wake them up. All functions are built in Alarm Fragment. It could work both offline and online. After user login, observer users are available to set up alarm for normal users. In the other words, the normal users could receive alarms from observer users.

3.1.2 Module 2 – Calendar module

The calendar module allows users to set up event for themselves to schedule their time. All functions are built in Calendar Fragment. Same as the alarms setting module, it could work both offline and online. Observer user could set event for normal users.

3.1.3 Module 3 – Personal module

Major personal module's functions are built in Me Fragment. This module allows users to register, login, and sign up. It is the window for users to choose offline or online services. Observer users could send relationship request to normal users from this module. After building relationship. Observer users could set up alarms or events for normal users via Alarms setting module and Calendar module.

3.1.4 Module 4 – Alarm activity module

The Alarm activity module will be activated when it is the time for the alarm clock. In this module, users will need to do the activity they choose to turn off the alarm ring.

3.2 COMPONENT -2 - BACKEND COMPONENT

The BackEnd Component has four major modules. User module, Alarm module, relationship module, and the Event module. These data are stored in MySQL, and then communicate with the client through spring boot

3.2.1 Module 1 – User Module

The purpose of this user table is to store the user's id, name, password, gender, email, type information. This information can help the client to authenticate the user information and confirm the user type

3.2.2 Module2 – Alarm Module

The purpose of this alarm table is to store the alarm information of each user. It includes user's id, time, repeat information, repeat day information and alarm type. This information allows the client to determine the id and time set by the user. The most important thing is that it also records the type of alarm clock, such as shaking the phone to release the alarm clock or completing the math problem to release the alarm clock.

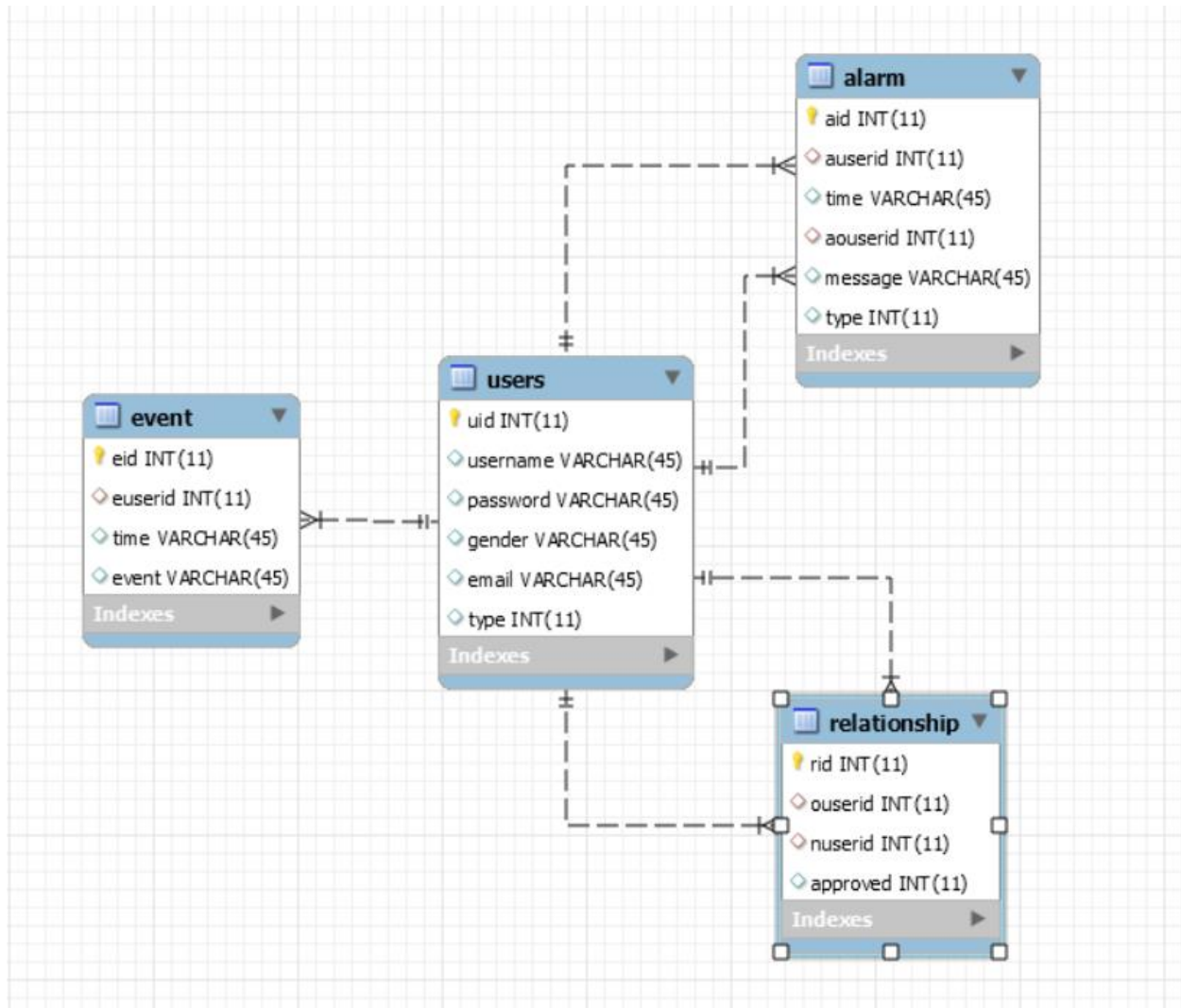
3.2.3 Module3 – relationship Module

This table stores the relationship between the Observer user and the normal user. It contains the Observer user id, normal user id, and whether the information is approved. This information allows the client to clearly know which two have relationship and which two are not, and also they can know which two relationship are pending.

3.2.4 Module4 – Event Module

This table stores the Event information about the client Calendar, which contains the user id, the time of the event and the event description. This information helps the client to understand the specific details of each event more clearly

4 Data Decomposition



5 Detailed Design

6 Design Rationale

6.1 DESIGN ISSUES

<ONLINE ALARM SAVING FAILED>

<IMPLEMENT CALENDAR VIEW AND ALERT DIALOG IN FRAGMENT >

6.2 <ONLINE ALARM SAVING FAILED>

6.2.1 Description

We are planning to do online alarms saving. However, the alarms save online would conflict with the local alarms. The alarms that set by observer users would be unavailable if the normal users did not receive it before the alarm time, but it still show activated.

6.2.2 Factors affecting Issue

- The format conflicts between local saves and online saves.
- The lack of real time update when the user is offline.
- Pending intent could not be set up when the user turns off the application.

6.2.3 Alternatives and their pros and cons

Alternatives-1:

We could give up the local saves and store every alarm online. In this way, there will no be any conflict between local saves and online saves.

Pros:

- No needs for local saves.
- Easier to regulate the data.

Cons:

- Online services only.

Alternatives-2:

We design two kind of alarms. They look the same from users' view. One is local alarm, which could only save in local storage. The second one is online alarm which is created by observer users only.

Pros:

- By designing two kind of alarms, no more conflict exists.
- Offline and Online function works properly.

Cons:

- User's local alarms will not be saved online, which means they could not get alarms from one device to another device.
- Alarm could not be set as repeatable.

6.2.4 Resolution of Issue

We choose the second solution since we do not want to give up offline functions. In fact, most of alarm application is not capable to move alarms from one device to another. We believe that giving up this function is the best way to solve the problem. Also, observer users could receive message when the normal users cancel the alarms they set up.

6.3 <IMPLEMENT CALENDAR VIEW AND ALERT DIALOG IN FRAGMENT >

6.3.1 Description

Because we are using the bottom navigation view to switch between tabs to let user do different thing. By doing so we will need to use fragments as different tabs instead of activities. Fragments have many differences to be used compare to activities. In this situation, when we were trying to implement calendar view and alert dialog in the fragment, we faced many issues.

6.3.2 Factors affecting Issue

There are many differences between the usage of fragment and activity. To implement something in fragment is not the same as in activity so that it need more time to learn and implement.

6.3.3 Alternatives and their pros and cons

Alternative-1:

We can change all fragments to activities.

Pros:

- Easy to implement things later.

Cons:

- It takes a lot of time to change all of our app's fragments to activities.
- We need to learn how to use activities to achieve bottom navigation.

Alternative-2:

Learn how to implement those things in fragment.

Pros:

- Do not need to change our app layout. It is much safer to do.

Cons:

- Too less tutorials online for fragments.

6.3.4 Resolution of Issue

We decided to learn how to implement everything in fragments instead of changing everything to activity so that we do not need to risk our entire app's functions. After a long time of searching and trying, we finally resolved our issue.