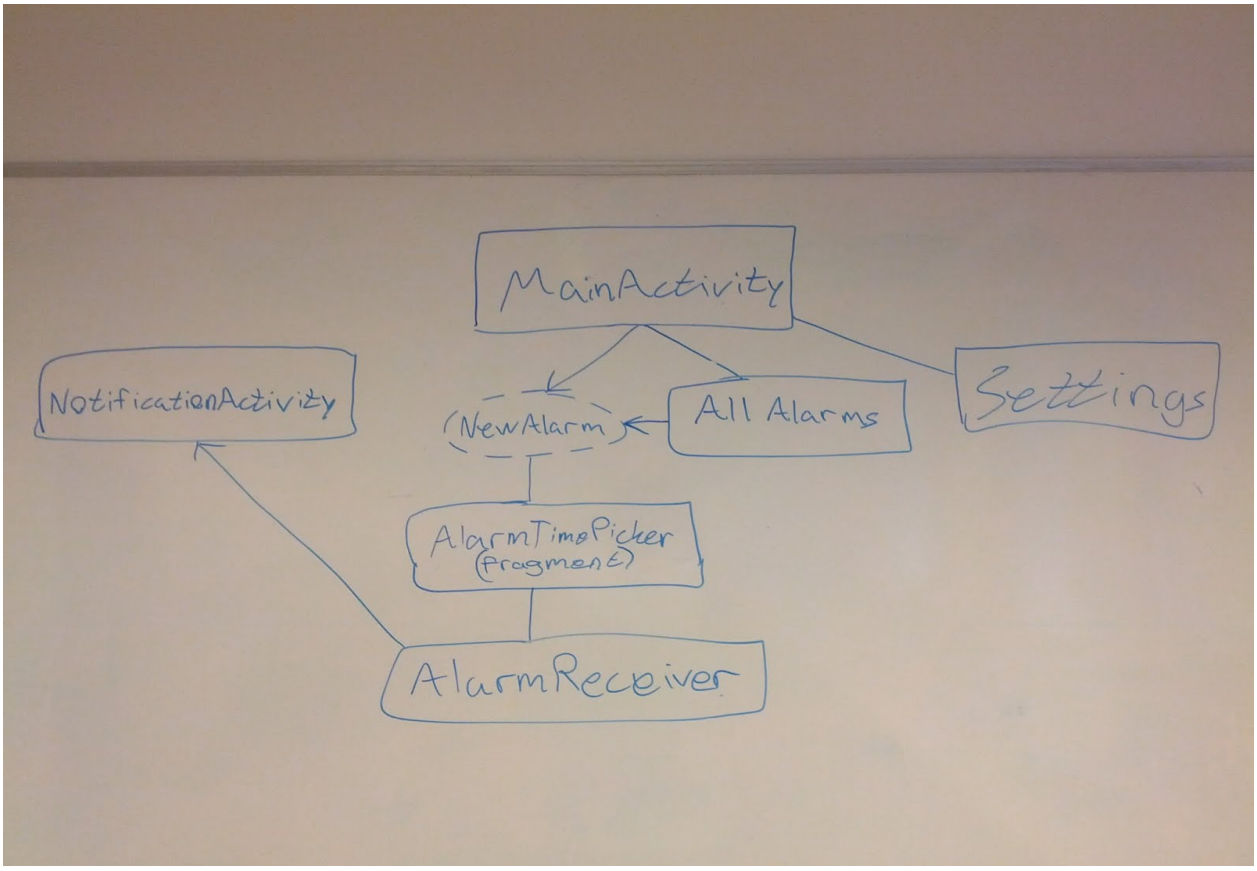
Code architecture



A somewhat outdated architecture is shown in the picture above.

1 - Architecture design

1.1 - Overview

We have a MainActivity that serves as a main hub of the program. From the MainActivity you can access all active alarms and also add a new alarm quickly.

When adding a new alarm the AddAlarmActivity will let the user choose the time which the alarm should go off. The activity will then add the alarm to the android AlarmManager which will take care of making sure the alarm goes off correctly. When the time has come the AlarmManager will send an notification to the AlarmReceiver which handles the alarm and acts accordingly. For now the AlarmReceiver will start the NotificationActivity which will show up on the users screen.

1.2 - Software decomposition

1.2.1 - General

The application will be divided into the following modules:
- Activities, various views with view-specific logic.
- Model

1.2.2 - Tiers

- Application
- Dataprovider
- Server

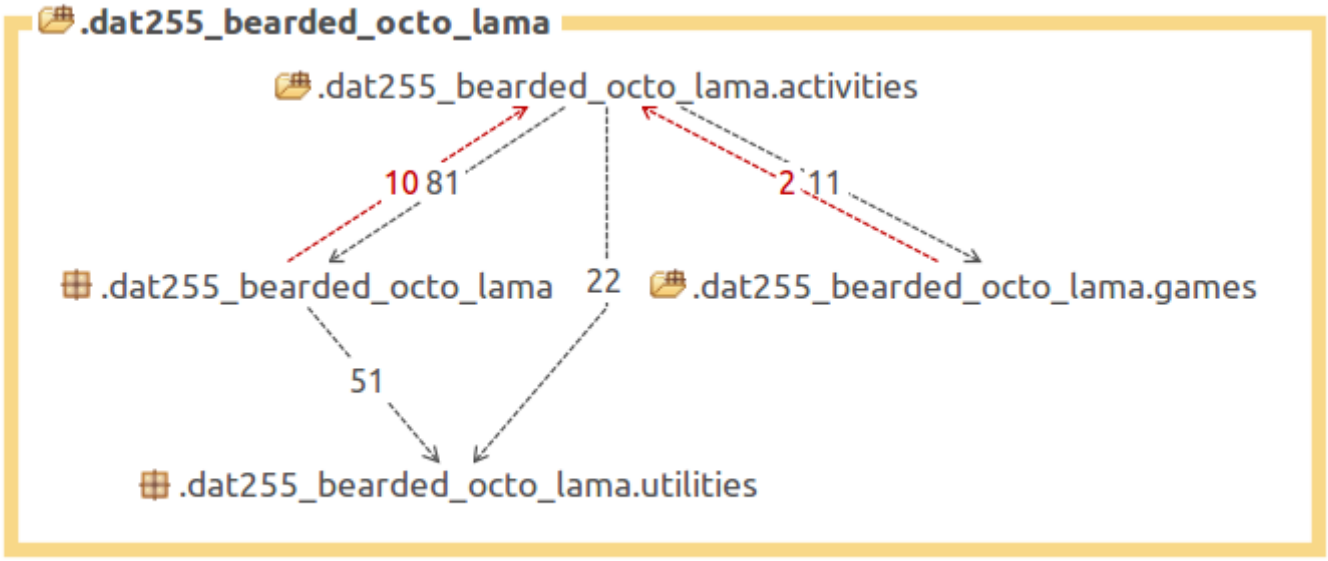
1.2.3 - Decomposition into subsystems

- All activities
- Data content provider

1.2.4 - Layering

N/A

1.2.5 Dependency Analysis



The reason we have circular dependencies is that our package structure is designed to be readable but not strictly code correct.

1.3 - Concurrency issues

N/A

1.4 - Persistent data management

Solved using a custom ContentProvider that uses the SqlLite3 database.

1.5 - Access control and security

N/A

1.6 - Boundary conditions

N/A