**Project 4 Task 3 – Age Calculator App**

**Description:**

The application gets date of birth from the user, and returns the person’s age.

1. Implement a native Android application

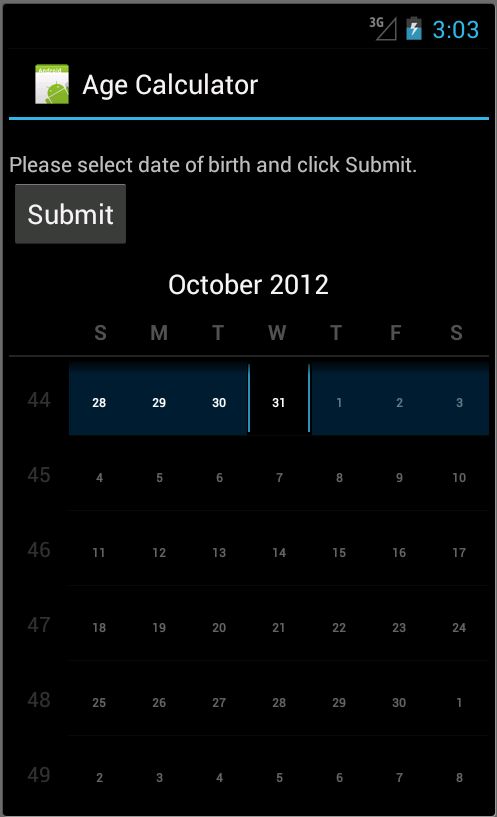
The name of my native Android application project in Eclipse is:

ds.project4task3

1.1. Has at least two different kinds of views in your Layout (TextView, EditText,

ImageView, etc.)

My application uses TextView, Button, and CalendarView. See main.xml for details of how they are incorporated into the LinearLayout. Here is a start-up screenshot



1.2. Requires input from the user

User can select date from calendar and click Submit button.

1.3. Makes an HTTP request (using an appropriate HTTP method) to your web app

My application does the HTTP request in GetPersonAge.java. The HTTP request is:

"http://www.project4task3-karinya.appspot.com/project4task3servlet?year=" + c[0].get(Calendar.*YEAR*) + "&month=" + c[0].get(Calendar.*MONTH*) + "&day=" + c[0].get(Calendar.*DATE*);

Where c[0].get(Calendar.*YEAR*), c[0].get(Calendar.*MONTH*), c[0].get(Calendar.*DATE*) are year, month, and date of birth, respectively.

The search method makes this request of my web application. The web application then return XML formatted of person age.

1.4. Receives and parses an XML formatted reply from the web app

An example of the XML reply is:

<getAgeResponse>

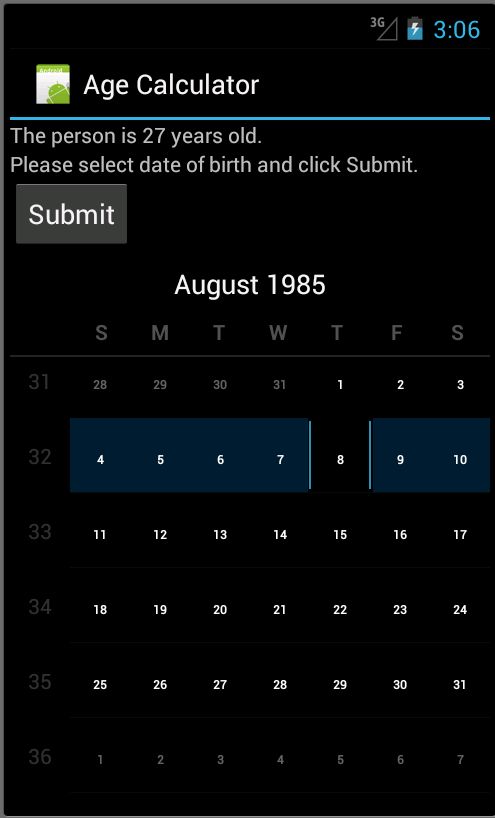
<Age>27</Age>

<Result>OK</Result>

</getAgeResponse>

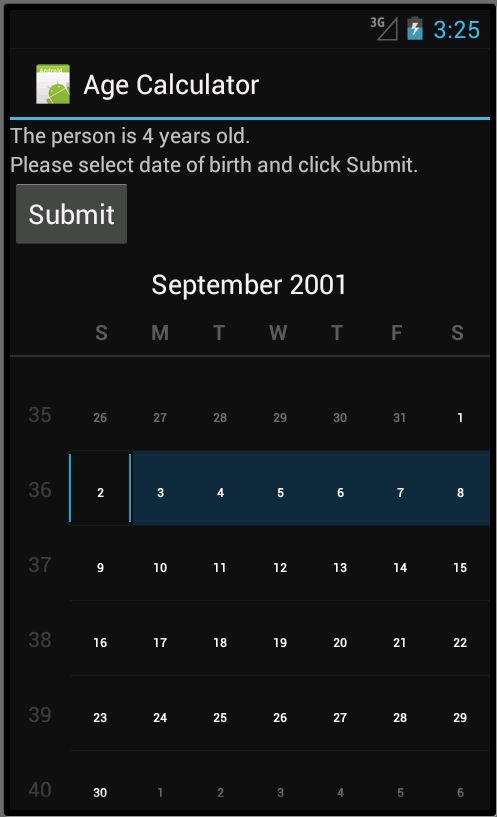
1.5. Displays new information to the user

Here is the screen shot after the picture has been returned. Age of a person is shown.



1.6. Is repeatable (I.e. the user can repeatedly reuse the application without restarting it.)

The user can select new date of birth and click Submit button. Below is the screen shot after user calculated younger brother age, now selected older brother birth date and is going to click submit button.



2. Implement a web application, deployed to Google App Engine

The name of the Google App Engine project in Eclipse is:

Project4Task3

2.1. Uses the MVC design pattern

In my web app project:

Model: AgeModel.java

View: age.jsp

Controller: project4task4servlet.java

2.2. Receives an HTTP request from the native Android application

project4task4servlet.java receives the HTTP request and passes date of birth to the model.

2.3. Executes business logic appropriate to your application

AgeModel.java calculates age from the given date of birth and returns age back to the servlet.

2.4. Replies to the Android application with an XML formatted response.

age.jsp formats the response to the mobile application in a simple XML format of my own design:

<?xml version=*"1.0"* encoding=*"ISO-8859-1"*?>

<%@ page contentType=*"text/xml;charset=ISO-8859-1"* %>

<getAgeResponse>

<Age><%= request.getAttribute("age")%></Age>

<Result><%= request.getAttribute("result")%></Result>

</getAgeResponse>