Health Reporter

Health Reporter lets health professionals record, evaluate, and report the results of a health appraisal. This system is quite large so I have divided it into 5 independent parts. Three of the parts need two OS versions. So in total, I am offering 8 projects for 8 different teams. The projects are fully independent. Your team will not depend on the other teams in any way.

When picking this project, please indicate which of the following 8 variants your team intends to undertake:

- Health Reporter Part 1 (Windows)
- Health Reporter Part 1 (Mac)
- Health Reporter Part 2 (Google app engine)
- Health Reporter Part 3 (Android)
- Health Reporter Part 3 (iOS)
- Health Reporter Part 4 (Windows)
- Health Reporter Part 4 (Mac)
- Health Reporter Part 5

Terms

You will be working on this project under the terms of the following IP assignment, which has been recommended by your lecturer: [http://www.docracy.com/6480/short-ip-assignment-agreement-for-internet-startup](http://www.docracy.com/6480/short-ip-assignment-agreement-for-internet-startup)

Contact

From 5-Oct to 30-Nov, I will meet with each team member, in person, once per week. In September, we can communicate by email and Skype.

In previous years, I have been the customer of over half a dozen projects in this course. I find that regular communication and
meetings are vital. In the past, teams that met with me often were successful; teams that did not, were not. There were no exceptions.

I would prefer to meet with you individually, or in pairs. Full-team meetings can be difficult to coordinate, and they tend to drag on. But more important, when I meet with one student, I can focus on that student’s questions and concerns, and get a better feel for their understanding of the project.

thomas@narbeshuber.com
skype: narbeshuber
**Part I. Clients, Tests and Test Results**

**Desktop Application (Windows/Mac)**

**Clients** are those whose health is being evaluated. **Tests** are the measurements, observations, and calculations that form a health appraisal. Users add, update, delete, and view their clients and tests.

---

**Example of a Client:**

- **First Name:** Robert
- **Last Name:** Smith
- **Group Name:** University of Tartu
- **Last Appraisal:** 05 Jul 2016
- **Age:** 38

**Example of a Test:**

- **Name:** Subscapular Skinfold
- **Units:** mm
- **Decimals:** 1
- **Ratings:**
  - Ages 0-29
  - 30-39
  - 40-49
  - 50-59
  - 60-69

<table>
<thead>
<tr>
<th>Name</th>
<th>Units</th>
<th>Decimals</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscapular Skinfold</td>
<td>mm</td>
<td>1</td>
<td>Needs Improvement</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fair</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Good</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Very Good</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Excellent</td>
</tr>
<tr>
<td>Sum of Five Skinfolds</td>
<td></td>
<td></td>
<td>maximum</td>
</tr>
</tbody>
</table>
When the user selects a client, a new screen slides up showing the client’s complete appraisal history (their **Test Results** over time). The user can then add new measurements and observations.

The software evaluates test scores by comparing them to established ranges, for age and gender. Ranges are defined in a test’s properties. Overall results for health categories, like Anthropometry or Musculoskeletal, are computed as well. Users can enter formulas to compute additional health indicators, such as: \( \text{BMI} = \frac{\text{client weight (kg)}}{\text{square of client height (cm)}} \).

**TOOLS**

*Windows*: Visual Studio IDE / **C#** development language

*Mac*: Xcode IDE / **Swift** development language
Part 2. Syncing and Statistics
Web Service (Google App Engine)

As client and appraisal data is recorded, it is stored in local (sqlite) databases. This is true for both for desktop and mobile Health Reporter apps. The web service makes it possible to share data between users within an organization, regardless of how or where that data was first entered.

The apps developed in parts 1 and 3 will use the web service’s API to upload/download the records for clients, tests and test results.

The second job of the web service is to build a worldwide store of health appraisal data, and to be able to answer the question:

“For my age and gender, how do I rank for test X in my organization, in my town, in my state, in my country, and in the world?”

For small datasets, the code required to answer this type of question is trivial. But for very large datasets, it becomes quite challenging. (Good luck!)

TOOLS
Google’s App Engine Launcher sets up a development server, and uploads files to App Engine. Choose your own IDE, and use either Go or Python as your development language.
Part 3. Clients and Test Results
Mobile Application (Android/iOS)

The mobile app is a companion to the desktop app. It lets users record test results more conveniently, at the time and place of the appraisal.

Like the desktop app, users can add, update, delete, and view their clients and test results, as well as evaluate scores and formulas. But the mobile app does not manage a database of appraisal tests. Users must first create their tests with the desktop app, and then sync them using the web service.

The mobile app also has a second purpose. It lets clients view their appraisal results on their phones, rather than in static documents.

<table>
<thead>
<tr>
<th>Robert Smith, 28</th>
<th>Overall</th>
<th>Triceps Skinfold</th>
<th>Biceps Skinfold</th>
<th>Waist Girth</th>
</tr>
</thead>
<tbody>
<tr>
<td>University of Tartu</td>
<td>77%</td>
<td>11.3 mm</td>
<td>10.3 mm</td>
<td>132.3 cm</td>
</tr>
<tr>
<td>Questionnaires</td>
<td>67%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anthropometry</td>
<td>80%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aerobic Capacity</td>
<td>37%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Body Composition
- Triceps Skinfold: 11.3 mm (13%)
- Biceps Skinfold: 10.3 mm (8%)
- Waist Girth: 132.3 cm (19%)

Percent Body Fat
- 12.5% (Optimal: 5.0 - 15.0%)

TOOLS
Android: Google App Studio IDE / Java development language
iOS: Xcode IDE / Swift development language
Part 4. PDF Reports
Desktop Application (Windows/Mac)

This software turns the appraisal results for a client into a PDF report. The report will include numbers, graphics and rich text.

How test results are displayed in a report is determined, in part, by each test’s properties. In addition, users can choose various report presentation options, like text size, rating symbol shape and color, horizontal or vertical bar charts, cover page style, and page orientation.

<table>
<thead>
<tr>
<th></th>
<th>13.07 2016</th>
<th>08.14 2015</th>
<th>22.12 2014</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Anthropometry</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Triceps Skinfold</td>
<td>mm</td>
<td>9.1</td>
<td>9.6</td>
</tr>
<tr>
<td>Biceps Skinfold</td>
<td>mm</td>
<td>8.2</td>
<td>8.4</td>
</tr>
<tr>
<td>Biceps Girth</td>
<td>cm</td>
<td>39.0</td>
<td>38.9</td>
</tr>
<tr>
<td>Flexed Biceps Girth</td>
<td>cm</td>
<td>44.1</td>
<td>42.1</td>
</tr>
</tbody>
</table>

**Body Composition**
- Excellent
- Very Good
- Good
- Fair
- Needs Improvement

**Sum of Five Skinfolds**

<table>
<thead>
<tr>
<th></th>
<th>23.3</th>
<th>37.1</th>
<th>58.6</th>
</tr>
</thead>
</table>

**TOOLS**
Windows: Visual Studio IDE / C# development language
Mac: Xcode IDE / Swift development language
Part 5. Questionnaires
Desktop/Web Application (Google App Engine)

Questionnaires require a small desktop app, a web service, and a web app. You can choose either Windows or Mac for your desktop app.

Users create their questionnaires with the desktop app, and then upload them through the web service.

<table>
<thead>
<tr>
<th>Questions</th>
<th>Responses</th>
<th>Guidance</th>
<th>Men</th>
<th>Women</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 1...</td>
<td>Response a to question 1...</td>
<td>Guidance for response a...</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Response b to question 1...</td>
<td>Guidance for response b...</td>
<td>2</td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td>Response c to question 1...</td>
<td>Guidance for response c...</td>
<td>3</td>
<td>2.5</td>
</tr>
</tbody>
</table>

The web app presents questionnaires to clients, and records their responses. Finally, the desktop app downloads the responses to the local database, and computes an overall questionnaire score for each client.

TOOLS
Windows: Visual Studio IDE / C# development language
Mac: Xcode IDE / Swift development language
Web: Google’s App Engine Launcher sets up a development server, and uploads files to App Engine. Choose your own IDE, and use either Go or Python as your development language.