Prototype

Design rationale

Choice of API-level
The group chose to have an API level of 15 and Android version 4.0.3, to be able to reach out to as many users possible but still have the opportunity to use as many features as possible. By choosing API level 15 the application reaches out to 98.4% of all Android users. However, an even lower API level could have been used since the different features introduced in API 15 wasn't used. But in the beginning of the team did not have any experience with Android development hence the API level.

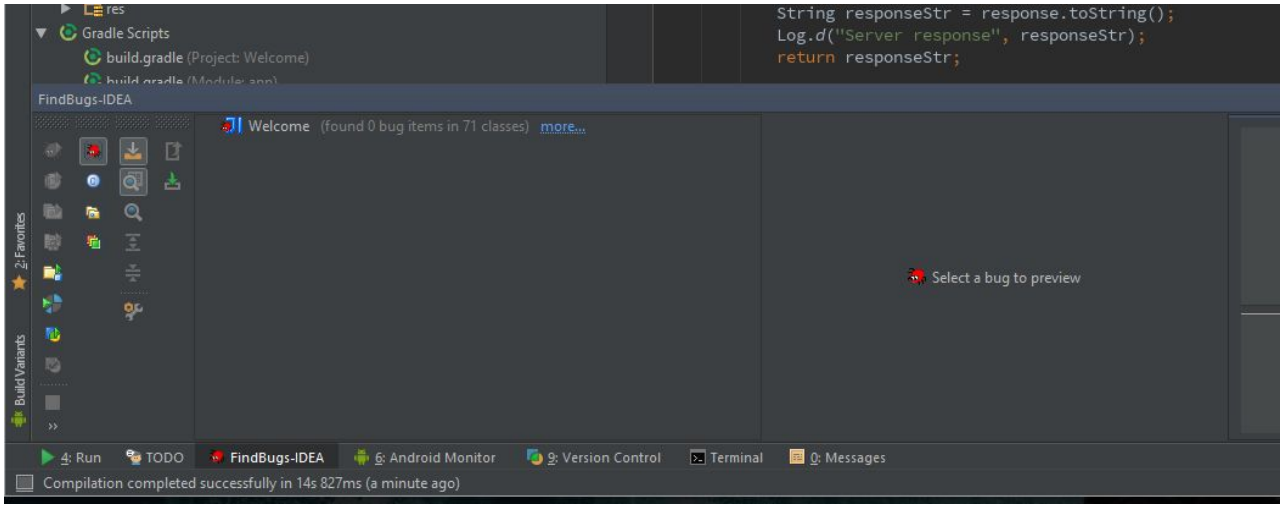
Database structure
The backend is built in Ruby on Rails and uses its standard database, SQLite. The database currently consists of three tables. One for users, containing all the data about a user such as name, date of birth, profession. Another one for matches between users where every entry contains the id of two users. Lastly one for all the messages where every message belongs to a match from the second table, has a sender id and a body for the text.

Application structure
The application requires an external server hosting the database to function, other than that there are no external dependencies.

The application has a relatively simple structure. Standard Android patterns are used with all UI code in one package. When switching between some activities, extra data is sent through intents. SharedPreferences are used for saving data long term, such as authentication token for the server, name and profession. This data is stored as key-value pairs. Keys for the SharedPreferences are saved in a class for ease of access. Code for sending and getting data from the server is in a package separate from the UI code. This is easily accessible through static methods which return a string which is then handled as either a JSON Object or a JSON Array depending on what data is returned.

The first time the application is started, the user goes through a setup wizard in the activities: LanguageActivity, MentorChoiceActivity, GenderAndBirthActivity and lastly JobActivity. Once the JobActivity is completed, a user is created in the database, data is saved in SharedPreferences and the user searches for a match. After the setup wizard is finished, the user is taken to MainActivity where the server is polled every other second, loading the user’s matches. Clicking on the name of a match in the list will take the user to the ChatActivity, which also polls the server for messages every other second. From the MainActivity the ProfileActivity is also reached where the user can change their profession, and interest.

Find bugs
All the bugs found by FindBugs have been fixed.



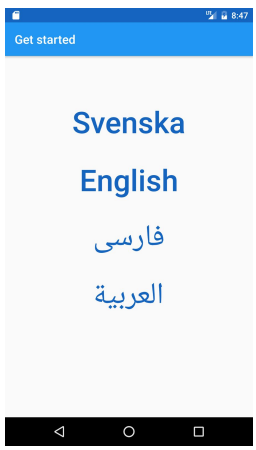
User stories
All of our user stories can be found on this Trello board:
https://trello.com/b/BSaZpcxZ/welcome

Testing
Read about testing in the report for *Connection*, under 4.5 Testing and for the review of testing under 7.5 Testing.

Overview

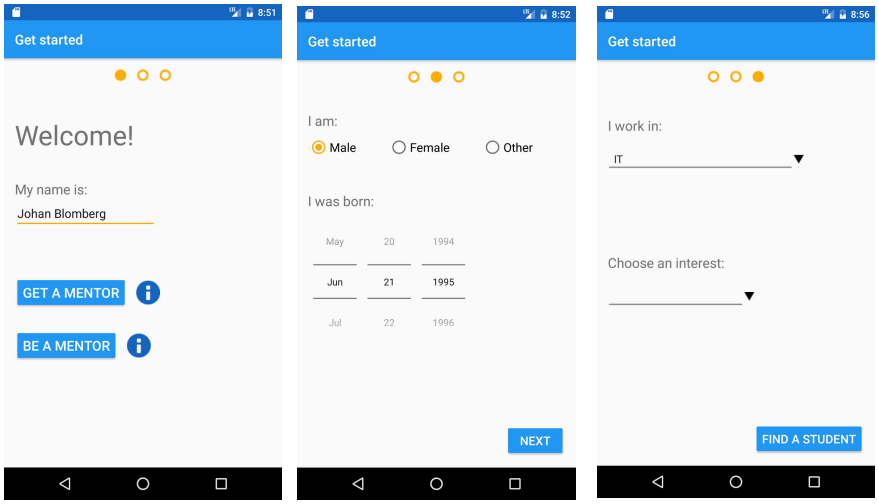
Structural and behavioural - Johan
The prototype consists of the following parts:

Language selection screen



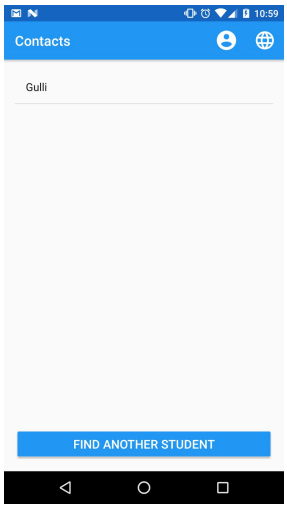
The first screen the user sees when starting the application for the first time. It presents a clear choice to the user about what language the interface should be in.

Profile setup wizard



Once the user has selected a language, they are faced with the wizard above, consisting of three screens in sequence, where they are asked to enter their personal information, whether or not they want to be a mentor or be taught by one, as well as their current occupation or the occupation they're interested in. Once this is finished, the user's profile is set up and the application starts finding a suitable match for them.

Main menu

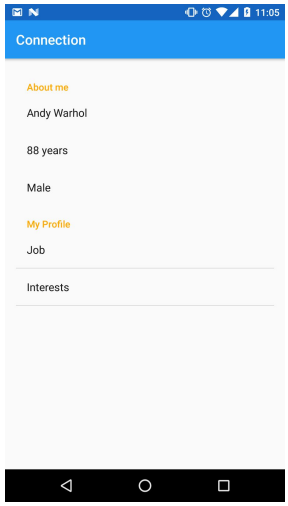


The Main menu is the screen a returning user sees when booting up the application, and the screen that new users are taken to once their profile is completed.
The bulk of this screen is occupied by the user's list of matches.

Pressing any name in this list will bring up the chat screen, where the user can chat with the selected mentor or student. Long-pressing a name will give the user the option to remove the selected person. The user can at any time press the button at the bottom of the screen to find another match, which will then be added to the list.

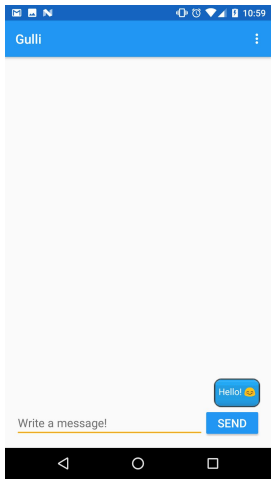
The buttons at the top lead to a profile screen and a dialog box for changing the application's language.

Profile



At the profile Screen, the user can look at their profile and change their selected occupation and interest.

Chat



From the chat screen, users communicate with each other via text messages. From the menu in the top-right, the user can remove this user from their list of contacts and prevent them from communicating with them.

Protocol (client/server) - John

The application uses HTTP requests, primarily GET with POST used to create users and send messages etc. The server returns JSON objects due to the ease of use in both Ruby on Rails and Android. Data returned from the server, such as lists of matches and messages, is then used in the application when rendering the list of matches or the chat.