Design Doc

Introduction

Scope

Looking for a babysitter for your kid can be a challenging task. Common solution is whatsapp groups specifically for this goal. However, when someone send a general message in this group, many times the replies are not relevant (for example, too young babysitters), or from unfamiliar babysitters, so one cannot be sure how reliable they are. In addition, in these whatsapp groups, babysitters and parents are constantly getting non-relevant notifications from other parents / babysitter.

This project suggests a new solution: social networks for parents to easily find a reliable babysitter for their kids. In the babysitter network each parent is connected to a group of babysitters he is familiar with. When parent wants to find one, he can send a message to the babysitters based on specific filtering parameters (for example: only above specific age or years of experience). This way the parents and babysitters get only relevant notifications, and parents can select only babysitter how is familiar to them.

Overview

This document describes the babysitter's social networks design and user interface:

1. Description of the features on the babysitter's social networks website.
2. Software architecture and technologies that will be used to build the website.
3. General description of the User interface.
4. Detailed description of the User interface, describing each screen, is on separate file: **screens.pdf**
5. List of databases and their content.
6. Tests and monitors.

Software description

General flow

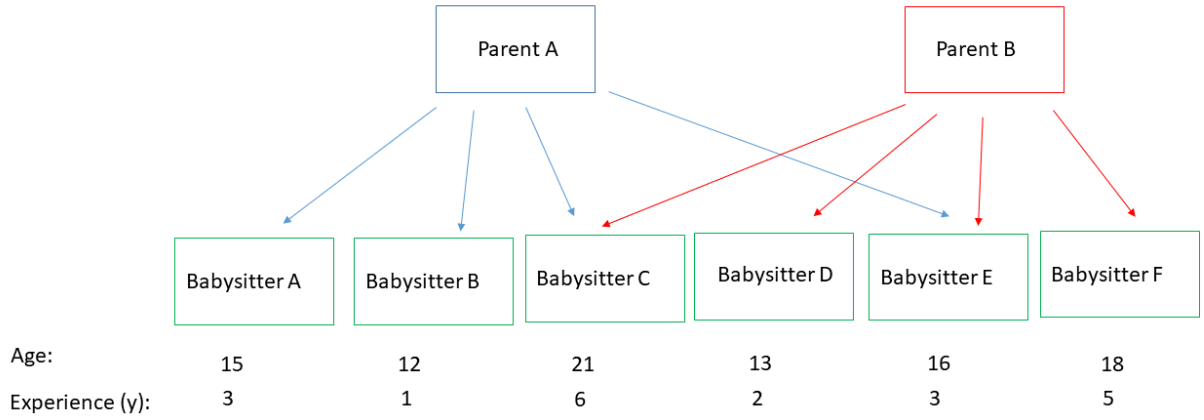
Network features:

1. Homepage.
2. Sing up page.
3. Login / Logout.
4. Settings page:
    a. For all users: user's details
    b. Only for babysitters: add information about age and experience
5. Add and remove connections:
    a. Selected from the list of registered users.
    b. Parents can add only babysitters and vice versa.
6. User's feed:
    a. Include the user information (Gender, age, etc.)
    b. Include the massages received and sent to other users.
7. Send messages to other users:
    a. Parents can send message to all connected babysitters according to selected parameters.
    b. Babysitters can reply to messages sent by parents.

Software architecture

The app will be built with Python-Django.
Database for users and connections will be stored using MySQL
Frond-end will be built using HTML, CSS and JavaScript.

Scheme describing the network connections:

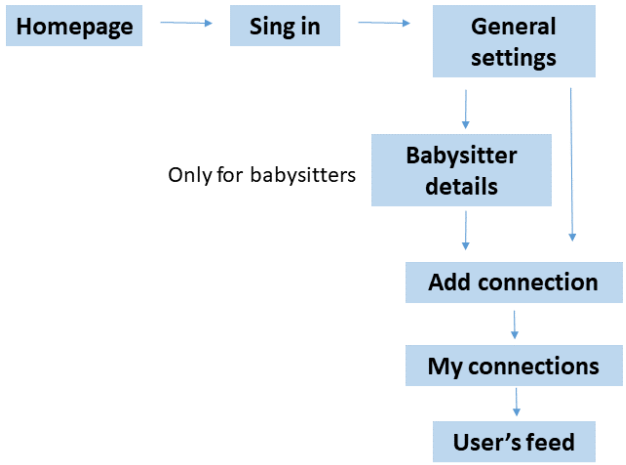


User interface

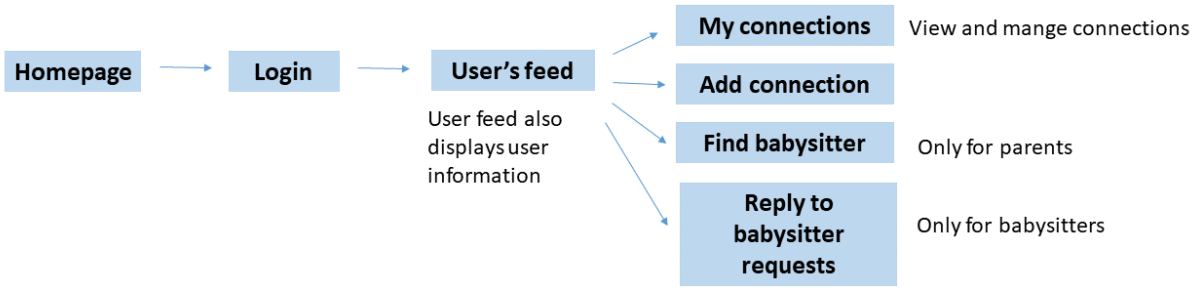
Detailed description of each screen in the website is in the attached file: **screens.pdf**

General flow is described here:

User's flow on the website – new user



User's flow on the website – registered user



Data handling

Database for users and connections will be stored using MySQL.

List of databases:

1. Database 1: Users Info
2. Database 2: Parents- babysitters Connections
3. Database 3: Babysitters details
4. Database 4: Parents- babysitter's messages
5. Database 5: babysitters reply messages

Database 1: Users Info

UserID	type	Name	Gender	email	Phone_number	image	slag
P-A	parent	Noam	M	Noam@gmail.com	052-5836328	Img File	userID
P-B	parent	Ruth	F	Ruth@gmail.com	053-7646328	Img File	userID
B-A	babysitter	Lily	F	Lily@gmail.com	052-5836780	Img File	userID
B-B	babysitter	Dan	M	Dan@gmail.com	052-5836355	Img File	userID
B-C	babysitter	Michal	F	Michal@gmail.com	052-5736325	Img File	userID
B-D	babysitter	Mona	F	Mona@gmail.com	057-9996324	Img File	userID
B-E	babysitter	Dana	F	Dana@gmail.com	054-6636328	Img File	userID
B-F	babysitter	Yael	F	Yael@gmail.com	050-7876328	Img File	userID

*Slag will be used to create users page (=feed)

Database 2: Parents- babysitters Connections

ParentID	BabysitterID
P-A	B-A
P-A	B-B
P-A	B-C
P-A	B-E
P-B	B-C
P-B	B-D
P-B	B-E
P-B	B-F

Database 3: Babysitters details

BabysitterID	Age	Experience
B-A	15	3
B-B	12	1
B-C	21	6
B-D	13	2
B-E	16	3
B-F	18	5

Database 4: Parents- babysitter's messages

massageID	ParentID	BabysitterID	content	Param_age	Param_exp	Param_gender
1	P-A	B-A	"today 2 h..."	13	2	F
1	P-A	B-C	"today 2 h..."	13	2	F
1	P-A	B-E	"today 2 h..."	13	2	F
2	P-A	B-C	"looking for.."	16	3	F,M
2	P-A	B-E	"looking for.."	16	3	F,M
3	P-B	B-C	"need babysitter..."	14	4	F
3	P-B	B-F	"need babysitter..."	14	4	F
4	P-B	B-C	"looking for tomorrow..."	15	3	F
4	P-B	B-E	"looking for tomorrow..."	15	3	F
4	P-B	B-F	"looking for tomorrow..."	15	3	F

Database 5: babysitters reply messages

massageID	BabysitterID	content
1	B-A	"I am available..."
1	B-C	"I am interested..."
2	B-E	"I can be there..."
3	B-F	"I am available..."
3	B-C	"I am interested..."
4	B-C	"I can be there..."
4	B-E	"I can come..."

Tests and monitors

Test

Website functions will be manually tested – add users, add connections, send messages and replies.

Logs

Django build-in logs will be used.