

FairPolitics - Fair feedback (SSD)

INTRODUCTION

Purpose:

The purpose of the design document is to provide a planning background for the organization. That is, how the system architecture will look, how to design the code structure, the structure of the classes, the data structures etc.

Scope:

The area in which our organization deals is improving the reflection of the will of the public.

Our goal is to provide a solution for elected officials to know in the best possible way what the public wants. Our system will provide an option for each selected public to post feedback for topics he or she is interested in focusing on, and the option to open a discussion for specific topics that the selected public or the public itself wants to bring up on the agenda.

Overview:

As part of the graduation project, our organization has chosen to focus on improving the reflection of the public will. The organization is divided into three roles for each member of the organization as follows:

Omer Shalom: Databases (MySQL)

Shai Bonfil: Back-End (NodeJS, Python)

Tal Schreiber: Front-End (ReactJS)

Reference Material:

During the project we will use several open source libraries to facilitate and expedite the product building process. The libraries we will use are:

1. Dynamic algorithm:

<https://git.tu-berlin.de/jonas.israel/dpr>

2. Convert a page to several languages:

<https://github.com/techomoro/ReactMultiPageWebsite>

3. Easy log in via facebook or gmail:

<https://github.com/zsasko/nodejs-social-auth-example>

SYSTEM OVERVIEW

In terms of functionality, the system will include the algorithms on which we are based, appropriate databases (MySQL), server management (rest API), clean and readable code that can be maintained with additional documentation, verbal description and examples of the various algorithms, service in a variety of languages, and UI / UX.

SYSTEM ARCHITECTURE

Architectural Design:

The structure of the code and the classes will be divided as follows in order to make order during the project in order to prevent trailing errors and future changes in some of the parts:

1. Login page: The login page for the system, after the login the user will reach the home page. In case the person does not yet have a user, he will go to the registration page.

2. Registration page: The page that each user will have to go through once in the first login, in order to exist in the system and then be transferred to the home page.

3. Home page: A page that shows all the users that the same user is connected to. This page contains all the entries to the profile page and advertising pages.

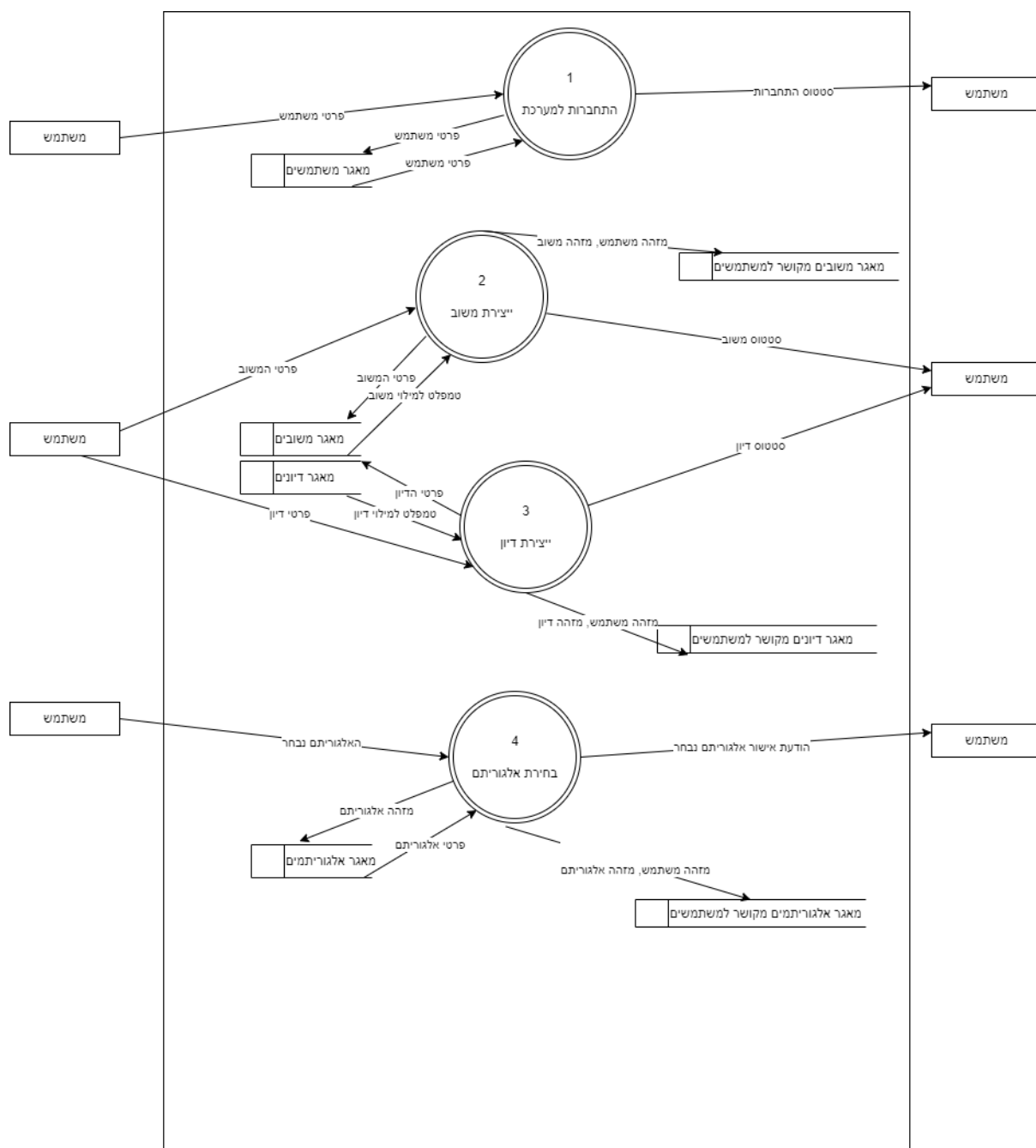
4. Profile page: Each user's private home page. This page contains the links to the home page and advertising pages.

5. Discussion post page: A discussion post page will contain an entry to the home page or profile, depending on where the entry was made respectively.

6. Feedback Posting Page: A feedback posting page will contain an entry to the home page or profile, depending on where the entry was made respectively.

7. The Algorithms Module: Will contain all the variety of different algorithms and links to the advertising pages and the profile page.

Decomposition Description: data flow diagram (DFD)



Design Rationale:

We chose the above module structure because it gives us the most solutions for our system requirements. This design provides low coupling and a high level of cohesion.

DATA DESIGN

Data Description:

We chose to work with a MySQL relational database. We will divide the tables as much as possible in order to reach the highest level of normalization as follows:

// Table of login details

1. Login table:

(id_user, phone_num, password) *optional - (facebook, gmail)*

// User information table

2. User information table:

(id_user, firstname, lastname, city, birthdate, job_title, description, profile_picture)

// A table of following and followers by id

3. Follower table:

(id_following, id_follower)

// Table of poll description

4. Poll table:

(id_poll, id_user, title, description, picture)

// Table of poll answer by poll id

5. Poll answer table:

(id_answer, id_poll, id_user, answer)

// The table indicates that a user voted for a particular answer

6. Poll answer approval:

(id_answer, id_user)

// Table of discussion description

7. Discussion table:

(id_post, id_user, title, tag, description, picture)

// Table of discussion comment by post id

8. Discussion responses table:

(id_post, id_user, id_comment, comment)

// Table that describe like to parent post by user id

9. Discussion like approval:

(id_post, id_user)

// Table that describe like to child comment by user id

10. Comment like approval:

(id_comment, id_user)

Data Dictionary:

Dynamic Proportional Rankings:

Proportional ranking rules aggregate approval-style preferences of agents into a collective ranking such that groups of agents with similar preferences are adequately represented. Motivated by the application of live Q&A platforms, where submitted questions need to be ranked based on the interests of the audience, we study a dynamic extension of the proportional rankings setting. In our setting, the goal is to maintain the proportionality of a ranking when alternatives (i.e., questions)—not necessarily from the top of the ranking—get selected sequentially. We propose generalizations of well-known aggregation rules to this setting and study their monotonicity and proportionality properties. We also evaluate the performance of these rules experimentally, using realistic probabilistic assumptions on the selection procedure.

Feedback:

This is a multi-choice survey where each user can mark several options. Feedback is raised by elected officials. The algorithm knows how to provide the maximum satisfaction of each user who responds to the survey, and not just displays data by absolute value of votes.

Discussion:

It's basically like a Facebook post so that anyone, elected officials and users, can write what they want. The algorithm knows how to present not only the discussions with the highest number of likes, but will also reflect the variety of topics that the public is interested in sharing.

COMPONENT DESIGN

Login:

When you want to connect to the system, the user is required to enter his phone number. The system will check that the phone number exists in the system, and verify that this number is indeed of the user that is trying to log in.

Registration:

In case a new user wants to log in, he is required to register. In this case the user will be required to enter a phone number, and the system will verify that the number does not already exist, so that users will not be able to skew results by opening multiple users. It will then send the user a temporary password that the user will need to enter to add another layer of security. The user will need to enter personal details such as first name, last name, city of residence, current role and profile picture. The more details the user enters, the higher the credibility percentage of the user in terms of the system.

Create feedback:

A feedback creation form is intended for the benefit of elected officials. An elected public that wants to post feedback will need to put a headline in the feedback. He can then enter as many answers as he is interested so that users can mark several options. In addition the selected public can add a more detailed description of the feedback, and also add photos if necessary.

Creating a discussion:

This option is designed to serve the public elected as well as the users. Anyone who wants to start a discussion can do so. Anyone who wants to start a discussion should first include a title for the discussion. You can then enter information and a full description of the discussion he wants to raise. It is also possible to add photos if necessary. Users who open a discussion can tag elected officials that they think the discussion is relevant to. In case there is an open discussion, there is an option to add a response to the discussion or to like the discussion.

Algorithm selection:

An elected public can choose which algorithm will process the results for feedback or discussions. He can go to a page where there is a literal description for each algorithm plus examples that illustrate how the algorithm works.

HUMAN INTERFACE DESIGN

Overview of User Interface:

Link to SRD that contains all system images

<https://github.com/Final-Project-bb/FairPolitics/blob/master/Documents/%D7%9E%D7%A1%D7%9E%D7%9A%20%D7%93%D7%A8%D7%99%D7%A9%D7%95%D7%A.pdf>