# OURGOVERNMENT

Requirements Document:

User Stories:

AS A/AN	I WANT TO…	SO THAT…
Politician (councilor in the Yavne municipality)	be able to understand the will of the citizens	So that I can serve the public in the best way
Israeli citizen	take part in the election of ministers	I feel there is a representation of my desires in government
Israeli citizen	change the way ministers are elected in government today	election of ministers in the government to be fair and not stipulated in coalition agreements
Politician (Spokeswoman for the Netanya Municipality)	be able to use an existing election results file and not have to manually enter all citizen elections into the app	My use of the app will be simple and easy
Israeli citizen	know how your algorithm works	I can be convinced it's fair as you said
Israeli citizen	be able to use the app for other purposes such as elections to university committees (student association)	I can use the algorithm for more fair distributions
Politician (councilor in the Yavne municipality)	be able to see voting statistics	I can know which groups in the population I am more / less valued

OURGOVERNMENT

Use Case:
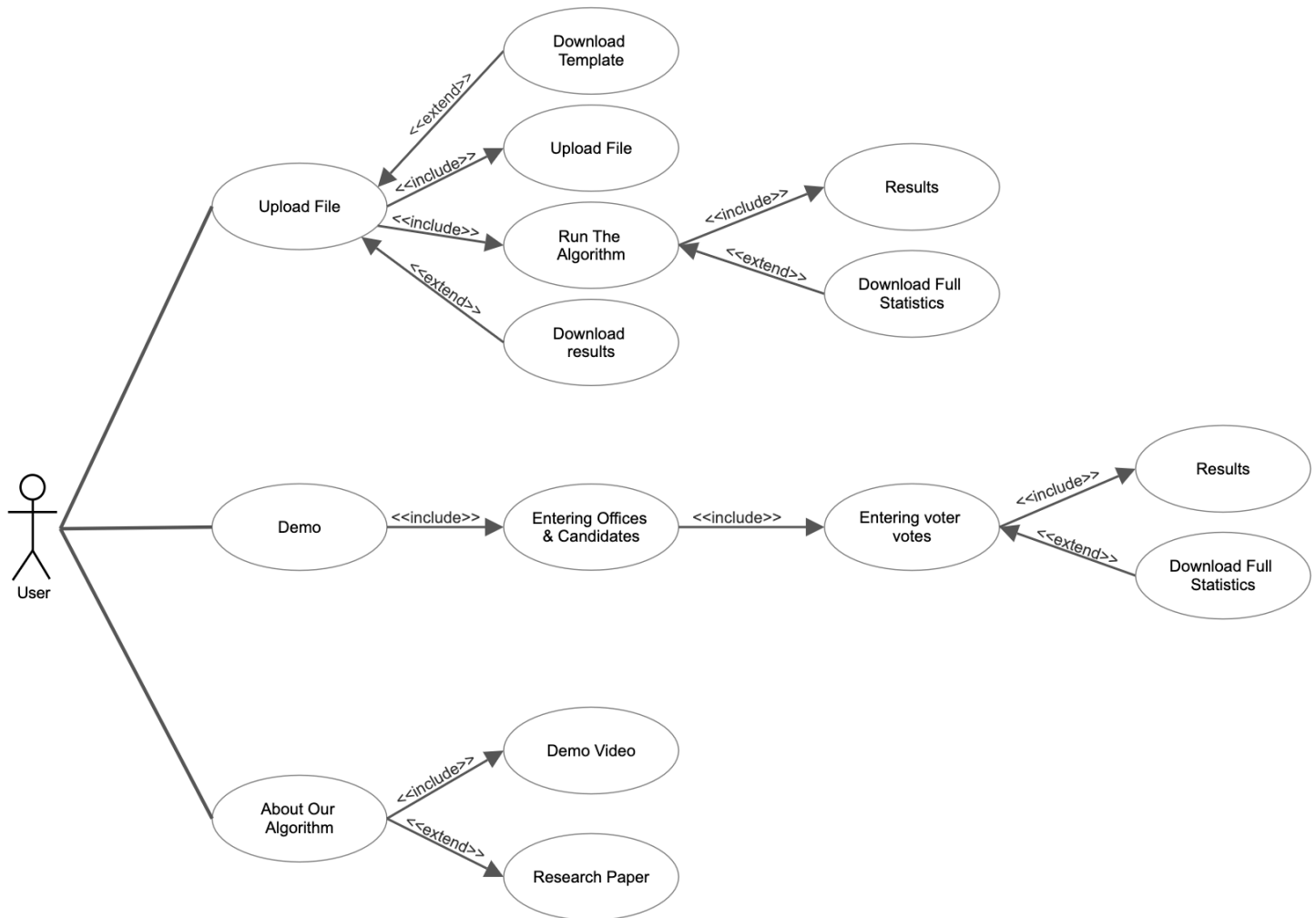
In the home screen the user can select several options:

1. <u>Upload a file</u> - In this screen the user can choose whether he wants to download the file template, upload a file, run the algorithm, see / download the results and download the full statistics of the algorithm run.

2. <u>Demo</u> - In this screen the user can enter in the application interface all the offices he wants to assign and all the candidates for each of the offices, in the next screen the user can enter the number of voters and their votes, in the next screen the user will receive the algorithm results (assignment of candidates to offices). In addition, if he wants, he can download the results and extended statistics.

3. <u>About the algorithm</u> - In this screen the user can watch a demo video for using the various functions of the system. In addition, he will be able to read the research paper we wrote about the algorithm with credit to an article by Rutvik Page, Ehud Shapiro and Nimrod Talmon.

Internal Operations:

In this section we will present for each scenario of using the system (Use Case) the collection of actions performed within the system, on the server side (which is not visible to the user).

Scenario 1: On the Home screen, the user clicks 'Upload File', downloaded the template file and uploaded it to the system, A check will be performed on the server side that the uploaded file is indeed in the appropriate format and that there are no errors in it (a check will be made that each candidate is running for at most one office and that the votes match the candidates), if not, an appropriate message will be displayed to the user.
If the format is correct and the user clicks 'Run the algorithm', the algorithm will run on the server side and output the result.
During the calculation a loading bar will be displayed to the user.
The result of the algorithm will then be converted to the system interface and displayed to the user with a detailed explanation of its correctness.
In addition, the algorithm generated 2 files in Excel format, a file containing the results of the algorithm run and a file containing additional statistics.
The user can choose whether to download one of the files by clicking 'Download results' or 'Download full statistics'.
* If the runtime of the algorithm is longer than usual, a window will be displayed for the user to fill in his email address and the results will be sent to the email box that the user entered.

Scenario 2: On the Home screen, the user clicks 'Demo', a screen will open for the user where he can fill in the names of the offices and candidates the user will be able to draw a random placement of offices and candidates by clicking on 'Random' - the system will draw offices and candidates from the current government.
In addition, the user will be able to enter the names of the offices and candidates himself - the data entered will be transferred to the database and from there as input to the algorithm in the form of lists (list of offices and list of candidates for each office).
At the end, the user will click on 'Next', a new screen will open where he will fill in the voters' votes - here too all the data entered will be transferred to the database and from there as input to the algorithm.
Then the user clicks 'Run the algorithm', the algorithm will run on the server side and output the result.
The result of the algorithm will then be converted to the system interface and displayed to the user.
In addition, the algorithm generated 2 files in Excel format, a file containing the results of the algorithm run and a file containing additional statistics.
The user can choose whether to download one of the files by clicking 'Download results' or 'Download full statistics'.

Scenario 3: On the Home screen, the user clicks 'About the algorithm', a screen will be displayed where he can watch a video explaining the use of the system and in addition the user will be able to read the research paper written about the algorithm.

Home Page:

http://www.ourgovernment.com

Search

OurGovernment

Ourovernment offers a fast and free solution for the selection of the executive branch, using methods that provide unquestionable fairness and are based on a unique algorithm.

Upload file

Download a template of the current government, enter all the voters' votes and let the algorithm provide you with a fair distribution of candidates for office

Start

Demo

Want to try the algorithm for yourself?
Want to make a fair distribution?
Create harmony by fair allocation.

Start

About our algorithm

Demo video for using the site And a comprehensive research paper on the GreedyPAV algorithm.

Learn more

OURGOVERNMENT

Upload File Page:

Title

http://www.company.com/upload

LOGO   COMPANY NAME   <u>Home</u>   <u>Products</u>   <u>Company</u>   <u>More</u>   Search 🔍

Electing the Executive Branch

Download Template   [Download]

Upload File   [Upload]

Run The Algorithm   [Run]

Download Results   [Results]

Statistics

[Download full statistics]

<u>Home</u> | <u>Products</u> | <u>Company</u> | <u>More</u>

OURGOVERNMENT

Demo Pages:

Title

http://www.ourgovernment.com/demo

LOGO | COMPANY NAME

Home | Products | Company | More | Search

Try Our Algorithm

The executive branch, or government, is typically not elected directly by the people, but rather formed by another elected body or person such as the parliament or the president. As a result, its members are not di- rectly accountable to the people, individually or as a group.

We consider a scenario in which the members of the government are elected directly by the people, and wish to achieve proportional representation while doing so.

We propose a formal model for allocation of k offices (ministries), each associated with a disjoint set of candidates contesting for that seat; a set of voters provide approval ballots for all offices.

We identify various scenarios pertaining to the 'power' exercised by different offices and propose axioms and algorithms to satisfy these axioms to determine pro-portionally representative office allocations. As using a simple majority vote for each office independently might result in disregarding minority preferences altogether, here we consider an adaptation of the greedy vari- ant of Proportional Approval Voting (GreedyPAV) to our setting, and demonstrate—through computer-based simulations—how voting for all offices together using this rule overcomes this weakness and satisfies a proportionality axiom. We note that the approach is applicable also to a party that employs direct democracy, where party members elect the party's representatives in a coalition government.

Office 1
(?) Ministry of Defence

Office 2
(?) Ministry of Health

Add office (+)

Candidates
(?) Beni Gantz

Candidates
(?) Nitzan Horowitz

Add Candidates (+)

(?) Clear

(?) Random

Next

Home | Products | Company | More

Title

http://www.ourgovernment.com/demo/next

LOGO | COMPANY NAME

Home | Products | Company | More | Search

Try Our Algorithm

The executive branch, or government, is typically not elected directly by the people, but rather formed by another elected body or person such as the parliament or the president. As a result, its members are not di- rectly accountable to the people, individually or as a group.

We consider a scenario in which the members of the government are elected directly by the people, and wish to achieve proportional representation while doing so.

We propose a formal model for allocation of k offices (ministries), each associated with a disjoint set of candidates contesting for that seat; a set of voters provide approval ballots for all offices.

We identify various scenarios pertaining to the 'power' exercised by different offices and propose axioms and algorithms to satisfy these axioms to determine pro-portionally representative office allocations. As using a simple majority vote for each office independently might result in disregarding minority preferences altogether, here we consider an adaptation of the greedy vari- ant of Proportional Approval Voting (GreedyPAV) to our setting, and demonstrate—through computer-based simulations—how voting for all offices together using this rule overcomes this weakness and satisfies a proportionality axiom. We note that the approach is applicable also to a party that employs direct democracy, where party members elect the party's representatives in a coalition government.

Number of voters 3

	Office 1	Office 2
#1 (?) Israel Israeli	Beni Gantz	Hili Tropper
#2 Alis	Naftali Bennett	Ayelet Shaked
#3 Bob	Gideon Sa'ar	Yoaz Hendel

(?) Clear

(?) Random

Run The Algorithm

Home | Products | Company | More

OURGOVERNMENT

About our algorithm Pages:

http://www.company.com/about

Search

About the algorithm

In the demo video, you can see how you can use the various functions in the application.

By uploading an Excel document in the format provided by us.

Or by manually entering the offices, candidates, and voting.

Credits to 'Electing the Executive Branch' wrote by Rutvik Page, Ehud Shapiro & Nimrod Talmon.

Research paper on the algorithm written by Itai Lashover, Liav Weiss, Amichai Kafka and Shani Levin

Research Paper

Home | Products | Company | More