Electronic Voting (e-voting)

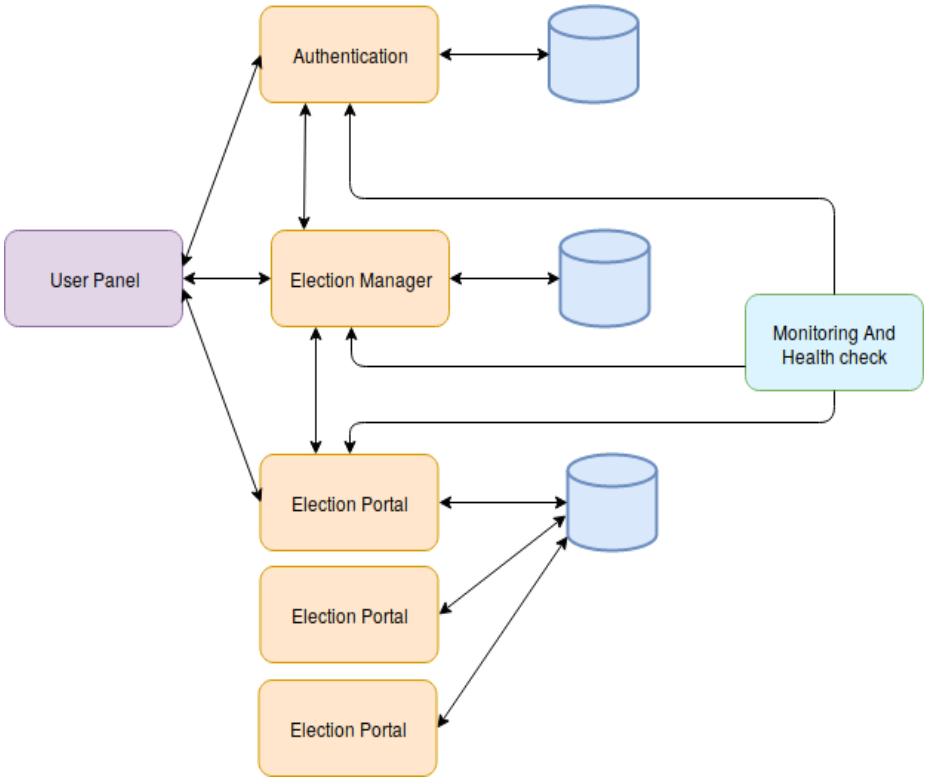
Cloud Computing Course Project, *Phase 0*
Submission Deadline: *Farvardin 21st, 1398*

Introduction

The push to online voting is growing worldwide every day. The benefits of e-voting are impressive and as a young or busy voter, it's the best easy way out! Vendors and governors have been looking for a fast, easy and low-cost way that provides maximum participation, for a long time. But, this is not the whole story! Unfortunately, downsides of this service are unacceptable yet. There is no such thing as a *secure* virtual voting system, despite what some vendors and governors might claim. But let's keep our fingers crossed that someday traditional voting methods are taken over by the e-voting. That being said, let's ensure that someday in the future, traditional voting methods are deprecate and e-voting celebrates the end of this battle.

Project Outline

The FUM Election System consists of four modules; *Election Manager*, *Election Portal*, *Authentication* and *User Panel*. Here is how these modules interact with each other.



Electronic Voting (e-voting)

Cloud Computing Course Project, *Phase 0*
Submission Deadline: *Farvardin 21st, 1398*

At this point, your assignment is to implement the Election Manager module. This Module is responsible for creating, Editing and removing elections (of course, removing elections is possible, only if the election is not running). ***It is your job to implement a service comprising of RESTful APIs over HTTP that fulfill the aforementioned tasks.***
An election model should *at least* have the following columns; **ID, Title, Start time, End time, listOfChoices, NumberOfVotes** (only number of votes submitted and not the model itself).
Note that, creation and editing of elections do not require authentication for now; however, in the next part of the project, your implementation should be modified to do so.

A summary of APIs to Implement are:

- void CreateElection(Election election)
- void EditElection(Election election)
- void RemoveElection(int electionId)
- void IncremenetNumberOfVotes(int electionId)
- List getListOfChoices(int electionId)
- List getAllElections()
- void electionExists(int electionId)
- String getElectionDetails(int electionId)

Feel Free to add other APIs if necessary.

Project Submission

All teams should create a private repository for their projects on GitHub and invite our accounts (*makbn*, *sayidhosseini*, *ardalanfp*) as collaborators to it. Each repository should have a Readme file that contains all necessary details about the service and all team members should change and commit the codes and files by their own GitHub accounts! Commits will be reviewed!