ZVotes by Zero Group - JavaEE Course 2014

Getting Started

Setup postgres (Optional)

- install postgres 9.3 link
- setup postgres

```
CREATE USER zvotes_db_user;
ALTER USER zvotes_db_user WITH PASSWORD 'zvotes_db_pass';
ALTER USER zvotes_db_user CREATEDB;
ALTER USER zvotes_db_user WITH SUPERUSER;
CREATE DATABASE zvotes_db_name;
GRANT ALL ON DATABASE zvotes_db_name TO zvotes_db_user;
```

Uni Koblenz VPN

Connect to university koblenz' VPN. For advice, see: http://www.uni-koblenz-landau.de/koblenz/GHRKO/netzwerk/vpn

Netbeans

- Install Netbeans with Java EE
- Open the Zero-Votes project with Netbeans
- Clean and build the Zero-Votes project
- Run ZVotes-web

Glassfish

Realm

- Open http://localhost:4848/
- Goto: Konfigurationen –> server-config –> Sicherheit –> Realms
- Create new one with:
  - name: `uniko-ldap-realm`
  - class-name: `com.sun.enterprise.security.auth.realm.ldap.LDAPRealm`
  - JAAS-Kontext: `ldapRealm`

    - Verzeichnis: `ldaps://ldap.uni-koblenz.de`
    - Basis-DN: `dc=uni-koblenz-landau,dc=de`
    - Gruppen zuweisen: `VALIDUSER`
- Add some additional Attributes:
    - search-filter: `(uid=%s)`
    - group-search-filter: `(member=%d)`
    - group-base-dn: `dc=uni-koblenz-landau,dc=de`

Mailsession

- Open http://localhost:4848/
- Goto: Ressourcen –> JavaMail-Sessions
- Create a new JavaMail-Session with:
    - JNDI Name : mail/uniko-mail
    - Mail Host : mailhost.uni-koblenz.de
    - Default User : username from uni-koblenz
    - Default Sender Address: address from uni-koblenz
    - Store Protocol : imap
    - Store Protocol Class : com.sun.mail.imap.IMAPStore
    - Transport Protocol : smtp
    - Transport Protocol Class : com.sun.mail.smtp.SMTPTransport
- Add some additional Attributes:
    - mail.smtp.password : Default User's password
    - mail.smtp.auth : true
    - mail.smtp.host : deliver.uni-koblenz.de
    - mail.smtp.ssl.enable : true
    - mail.debug : false

System vision

The Votes!-System supports electronic polls. The system shall be used mainly by university bodies such as the examination office (Prüfungsausschuss). Many

decisions in such bodies are made by polls. Since a common meeting is often hard to organize due to crowded schedules of participants, a voting system accessible via the Web is desirable.

Any member of the university must be able to define and conduct a voting. This role shall be called organizer. Participants, i.e. the people making a choice,need not be member of the university.

The system shall support different types of polls, such as yes-no, one of n options, m of n options.

Voters must be able to abstain from voting (vote void, ungültige Stimme oder Enthaltung). Depending on the decision mode (absolute majority relative majority, simple majority), void votes are counted differently.

Participants of a voting get a token, e.g. a transaction number, via E-Mail. This token represents their ballot paper (Wahlschein). The system must ensure anonymity. At no point in time, the person submitting a ballot must be connectable to her voting.

However, some administrative procedures require that the system keeps track of who has voted and who didn't participate. The organizer can decide if such tracking is required. Optionally, the voting system can send reminder mails to voters who did not yet participate.

Once a voting has been started, nobody can change the options and the participant list any more. After the voting deadline expires, the organizer can view the results. To ensure anonymity, the results may not be shown when an identification of a participant is possible (e.g. when tracking is enabled, and only one participant submitted her decision).

A graphical representation of the results, e.g. as a bar chart or pie chart, would be nice. For subsequent polls, an organizer should be able to record participant lists. The lists can easily be reused when the same participants occur many times, a usual situation in university bodies.

Fulfilled requirements

1. **Polls**

   1.1 The system must support electronic polls with one or more items. (*Set*)

   1.2 .Each poll must have a title. The title has to be unique (scope is system). (*Controller validated and unique fields*)

1.3 Each poll must have a description. (*It is displayed on the voting page, right below the poll name*)

1.4 Each poll must have a voting period (start and end date with time). (*When a poll is published, it will be started at the start time via an asynchronous task. It will be also finished via such a task.*)

1.5 Each poll must have at least one item. (*Tested before publishing*)

1.6 The system must allow to group arbitrary many items into one poll. (*Done*)

2. **Poll states**

   2.1 The system must implement poll states. Polls be in one of four states: PREPARED, STARTED, VOTING, FINISHED. State changes are specified below. (*When a poll is created, its state is PREPARING. When it is published it will be PUBLISHED. At the start time it's set to STARTED. After the first vote it's VOTING. At the end time it is set to FINISHED.*)

   2.2 When all participants submitted their votes, the system must set the poll's state to FINISHED. (*Done*)

   2.3 An organizer must be able to extend the voting period when a poll is in state STARTED or VOTING. (*Done*)

3. **Organizers**

   3.1 The system must allow all university members to act as organizers. (*Every university member can login via LDAP with username and password and automatically has a organizer instance.*)

   3.2 Organizers must identify themselves with username and password. (*see 3.1*)

   3.2.1 University members can be identified by the LDAP service provided by the GHRKO computing center. (*see 3.1*)

   3.2.2 If LDAP is not used, an administrator must be able to create organizer accounts. (*The system uses LDAP for login.*)

   3.3 An organizer must be able to conduct arbitrary many polls. (*Done*)

   3.4 The system shall provide a preview mode that allows organizers to view how their polls look like when a participant fills in her choices. (*The organizer is able to see a preview of the poll. This is possible via the preview button in the*

account panel.)

3.5 The system can provide a possibility to add further organizers to a poll. Each organizer must have the same options. (*An organizer can add other organizers to a poll.*)

3.5.1 An organizer must be able to change the organizer list (*Working, see 3.5*)

3.5.2 An organizer must not be able to remove herself from the organizer list. (*It is not possible for a organizer to remove herself from a poll.*)

4. **Administrators**

4.1 The system must support an administrator role. (*The system provides an administrator role. The first user, which logs in to the system, is the administrator*)

4.2 Administrators must have the ability to delete polls (including all votes) from the system. (*The administrator has the ability to delete polls.*)

4.3 Administrators must not be able to view votes nor results of polls which they didn't organize. (*The administrator is not able to see votes or results of polls.*)

4.4 Administrators must be able to create and delete user accounts, if no LDAP authentication is provided (see requirement XXX). (*Using LDAP, see 3.1*)

5. **Participants**

5.1 The organizer of a poll must be able to invite 3 to arbitrary many participants. In polls with less than 3 participants, anonymity can not be asserted. (*An organizer invite as many participants as he like but cannot publish a poll with less than 3 participants.*)

5.2 Each participant must be identified by her email address. (*Each participant is identified by her email address.*)

5.3 The system shall support participants from outside the university. (*As long as the participant has a valid email it doesn't matter if he belongs to the uni koblenz.*)

5.4 After a poll is STARTED by the organizer, each participant must be informed via email about the poll. (*Done via Java-Mail, sending real mails*)

5.5 The information mail must include the title of the poll, the start and end

dates, the number of participants, and a token. (*Done via email templates in locales*)

5.6 The information can also contain a hyperlink immediately referring to the voting page with a pre-filled token field. (*Working as well via url/token/tokenstring*)

6. **Participant lists**

   6.1 The organizer must be able to modify the participant list until a poll is STARTED. (*The organizer is be able to modify the participant list until the poll is PUBLISHED.*)

   6.2 The system can provide a means to comfortably create participant lists, e.g. by pasting email addresses from other applications such as spread sheets or KLIPS. (*The organizer can create recipient lists and dump in emails in whatever format he likes. Emails get parsed an will be instantiated as unique recipient entries.*)

   6.3 The system shall provide a means to store participant lists for easy reuse in subsequent polls. (*An organizer can store recipient in recipient lists for an easy reuse of participant lists.*)

   6.3.1 The stored participant lists must be private to each organizer. (*The recipient lists are private for each organizer.*)

   6.3.2 Each stored participant list must have a unique name (scope is per organizer). (*Each recipient list has a unique title for each organizer.*)

7. **Tokens**

   7.1 The token must be randomly chosen. (*The token is generated by uuid.*)

   7.2 The token must be unique (scope is system). (*Each token is unique in the system.*)

   7.3 The token must be long enough to make it very very improbable that anybody can forge a valid token. (*The token is a uuid with a length of 32. It's very improbable to forge a valid token.*)

8. **Anonymity**

   8.1 The system must ensure anonymity. (*The system ensures anonymity as good as it is possible.*)

8.2 At any point in time it must be impossible to identify which participant submitted which vote. This also must be guaranteed for polls with participation tracking. (*As long as the database is not logged via an database-administrator it is not possible to identify which participant submitted which vote.*)

8.3 The system must ensure that a token can not be associated with a vote. (*As long as the database is not logged via an database-administrator it is not possible to identify which token was used for which vote.*)

9. **Participation tracking**

   9.1 The system must provide an option to enable participation tracking for a poll. (*An organizer can activate participation tracking per poll.*)

   9.2 The system can provide an option to configure automatic reminders via E-Mail. (*If participation tracking and send reminder mail are enabled per poll a reminder email will be send to participants, who did not vote until this time.*)

10. **Submitting a vote**

    10.1 The system must provide a web page to submit a vote. (*The system provides a website to enter a token, and another to submit a vote. This ensures that no person can see polls, items and options which should not be public.*)

    10.2 The voting page must present an input field for a participant's token. (*See 10.1*)

    10.3 The token input field can be pre-filled (see requirement XX). (*The token field can be pre-filled via an URL. The history stack gets changed via Javascript to ensure it's not possible for other users to use browser functionality like back or forth to get the user's token.*)

    10.4 After the token was verified, the system must display the items. (*After the token was verified the system displays the poll, items and options.*)

    10.5 The system must present a button to submit a vote. (*On the bottom of a page is a button to submit, cancel and abstain from all items.*)

    10.6 After a vote was submitted, the token used for that vote must be invalidated (*i.e. it can't be re-used, participants can not change their vote after submitting). (After the vote the token is marked as used. It's not possible to change a vote or reuse the token.*)

    10.7 The system must allow to cancel a voting (e.g. by closing the browser, or by clicking a cancel button). (*working cancel button*)

10.8 The token used in cancelled voting must be reusable later. (*The token is reusable if no vote was submitted.*)

10.9 For a cancelled voting, the system must not remember any of the choices. (*The system ensures that it's not possible to recover previous choices.*)

10.10 The system shall ensure that subsequent participants using the same browser window can not restore the previous choice (e.g. by the „go back" function or by auto fill capabilities of browsers). (*Working, see 10.9*)

11. **Abstain from voting**

    11.1 The system must provide a means to abstain from voting (Enthaltung oder ungültige Stimme) for each item of a poll. (*It's possible to abstain from a specific item, displayed as option, or to abstain from all items, displayed as button at the bottom of the voting page.*)

    11.2 The system can provide a means to abstain from voting for a complete poll (this is equivalent to abstaining from all items). (*See 11.1*)

12. **Types of items**

    12.1 The system must support different types of items. (*Each item has one of the types "YES/NO", "1 of N" and "M of N".*)

    12.2 The items of a poll must have a title (titles have to be unique with scope poll). (*Each item has a unique title in the scope of a poll.*)

    12.3 The options of an item must have a short name and a description. (*Each option has a short name and a description.*)

    12.4 The system must support YES/NO items. In this case, the short names of the options are "yes" and "no". (*"YES/NO" items has yes and no options.*)

    12.5 The system must support YES/NO items. In this case, the short names of the options are "yes" and "no". (*"1 of N" items give participants the possibility to vote for 1 of two or more options.*)

    12.6 The system must support M OF N items. Participants can choose at most M of two or more options (M≤N). (*"M of N" items give participants the possibility to vote for M of M or more options.*)

    12.7 The system can support 1 OF N items with a free text option. Participants can choose one of the predefined options. Alternatively, they can enter a free text to indicate their choice. (*The system supports free text options at "1 of N"*

or “M of N” items.)

12.8 The system can support M OF N items with up to M free text options. Participants can choose at most M of the predefined options. Alternatively, they can can use less than M predefined options an enter the rest of their choices into free text fields. (*At “M of N” items, the participant can choose 1 to M predefined options and enter 1 to M own options. Only 1 to M options overall are legit.*)

13. **Results**

    13.1 An organizer must be able to view the results of a poll after the voting period is FINISHED. (*An organizer is able to view the results after the poll is finished.*)

    13.2 Nobody must be able to view (intermediate) results in the STARTED and VOTING states. (*Nobody is able to see the results of a poll until the poll is finished.*)

    13.3 The system shall provide a means to publish results (e.g. by sharing a hyperlink). (*The system provides means to publish results by sharing a link*)

    13.4 The system must not show results of polls with less than 3 submitted votes. (*The system does not show results of polls with less than 3 submitted votes.*)

    13.5 The results view must show the number of votes for each option of each item of a poll, as well as the number of abstentions. (*The system shows the amount of votes for each option and the amount of abstentions for each item.*)

    13.6 The system can display the results in graphical form (e.g., as pie chart). (*The system displays the results as bar diagrams, using bootstrap-progressbars.*)

    13.7 The system prints out wether an option is winner due to relative, simple or absolute majority

14. **User interface**

    14.1 The system can use third-party CSS libraries (such as Twitter Bootstrap) to achieve a modern look and feel. (*The system uses Twitter Bootstrap, Bootstrap-Progressbar, Toastr (used for messages), Boostrap-Datetimepicker, Font-Awesome (Icons)*)

    14.2 The system can use advanced technologies (e.g. AJAX, JavaScript) to

enhance user experience. (*The system uses a few Javascripts for a better user experience.*)

14.3 The system can use third-party JSF components (e.g. myFaces, richFaces). (*The system uses a technology for serving nicer URLs, calls Rewrite. It enables URL-changing e.g it changes account/poll_list.xhtml to /account/polls/*)

14.4 The system must provide a user interface suitable for desktop/laptop browsers. (*Using Twitter Bootstrap, see 14.1, the web page uses a responsive design.*)

14.5 The user interface can support multiple devices (desktop, tablet, mobile, etc.). (*See 14.4*)

14.6 The user interface must be realized with JSF. (*The user interface is realized with JSF.*)

15. **Security, encrypted communication**

    15.1 The system must store passwords in encrypted form (unless LDAP is used for identification). (*We are convinced that the storage of encrypted passwords is generally unsafe because if the secret key gets leaked - all passwords can be decrypted so we choose to hash all passes (with a secret hash).*)

    15.2 All communication of the voting system with its users (administrators OPTIONAL, organizers, participants) shall be encrypted (HTTPS protocol). (*We are using the RedirectToHttpsFilter on /*. To include the welcome-file we are using a WelcomeServlet to redirect von / to the actual welcome file.*)

    15.3 The communication of the voting system with the database server can be encrypted. (*not implemented*)

16. **Internationalization**

    16.1 The voting system must provide a user interface in one of the two languages German or English. (*ZVotes-web/src/main/java/com/zero/votes/beans/LanguageBean, Languages are represented in a ResourceBundle in ZVotes-web/src/main/resources/com/zero/votes/Locale_LANGUAGE_CODE in Template activated via: <f:view locale="#{language.localeCode}"/>, plus entries in WEB-INF/faces-config.xml, Internationalized Strings available via #{msg.STRINGNAME}*)

    16.2 The voting system can provide a means to change the user interface

language. (*Available in Footer via LanuageBean and Drop-Down to Select.*)

16.3 The language switch can be done by detecting the client browser language settings. (*FacesContext.getCurrentInstance().getExternalContext().getRequestLocale();*)

16.4 The language switch can be done by clicks to icons or links. (*The language switch is realized with a form select in the footer on the right side which autosubmits itself on selection change.*)

17. **Browser support**

    17.1 .The system must support at least one of the following web browsers: FireFox, Safari, Chrome in the most recent stable version available at delivery of the project. (*The system is optimized for the current Chrome and Firefox browsers.*)

Glossary

A…

administrator : a user role in the system. Users with this role can create organizer accounts.

abstain (from voting) : Participants must have the option to abstain from voting (sich enthalten) for each item of a poll, or for a complete poll. When a participant chooses to abstain, her decision may influence the result of a poll, depending on the decision mode. When a participant chooses to abstain, her vote is taken into account when computing the voter participation value (Wahlbeteiligung).

D…

decision mode : The decision mode of a poll can be one of three modes: absolute majority: an option is accepted if more than 50% of the participants voted for that option relative majority: an option is accepted if more than 50% of the votes indicate that option simple majority: an option is accepted if it has more votes than all others

I…

item (of a poll) : TODO

O…

option (of an item) : TODO

organizer : TODO

P…

participant: TODO

participation tracking : TODO

poll : TODO

T…

token : TODO

V…

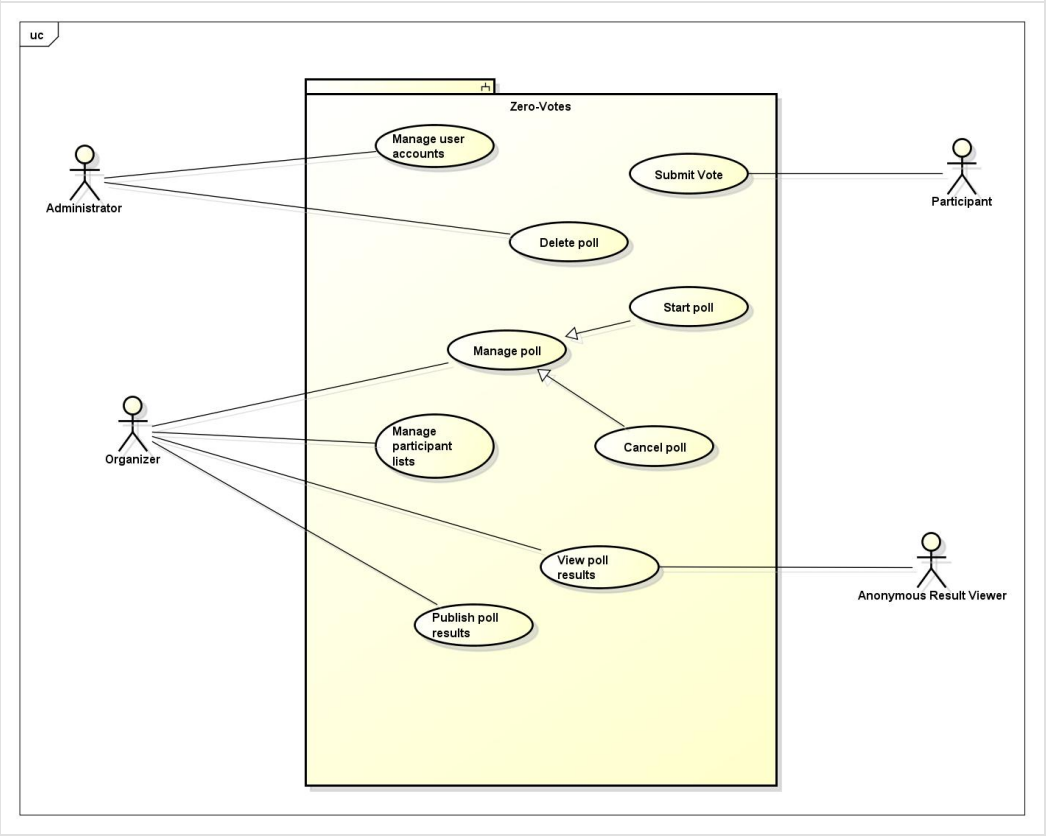
void(vote) : same as abstain

vote : TODO

voter : A participant who submitted a vote.

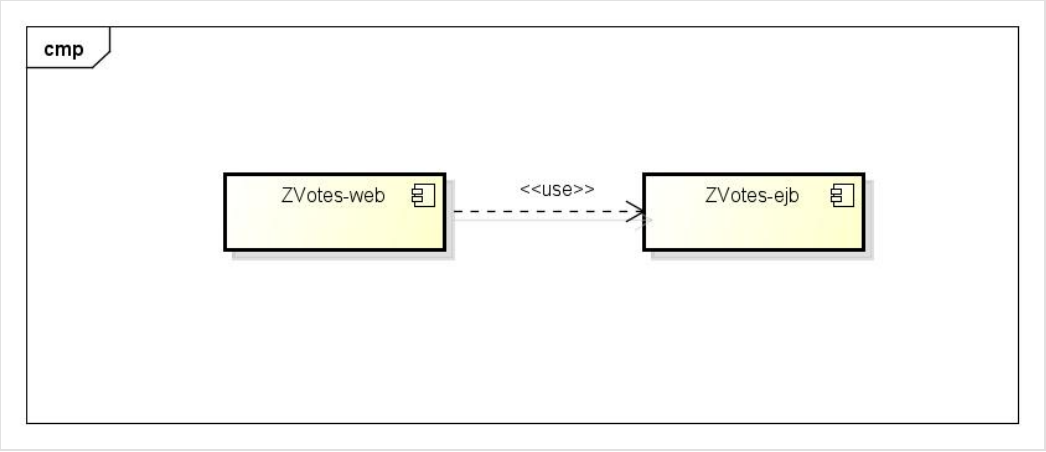
voter participation : The voter participation (Wahlbeteiligung) is the ratio of total number of voters to total number of participants. The voter participation has to be computed for each item of a poll.

Scenarios



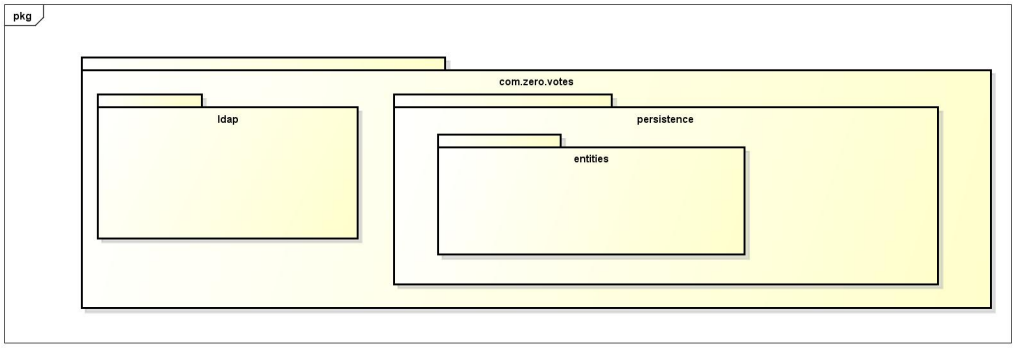
Use Case diagram not working

Project structure

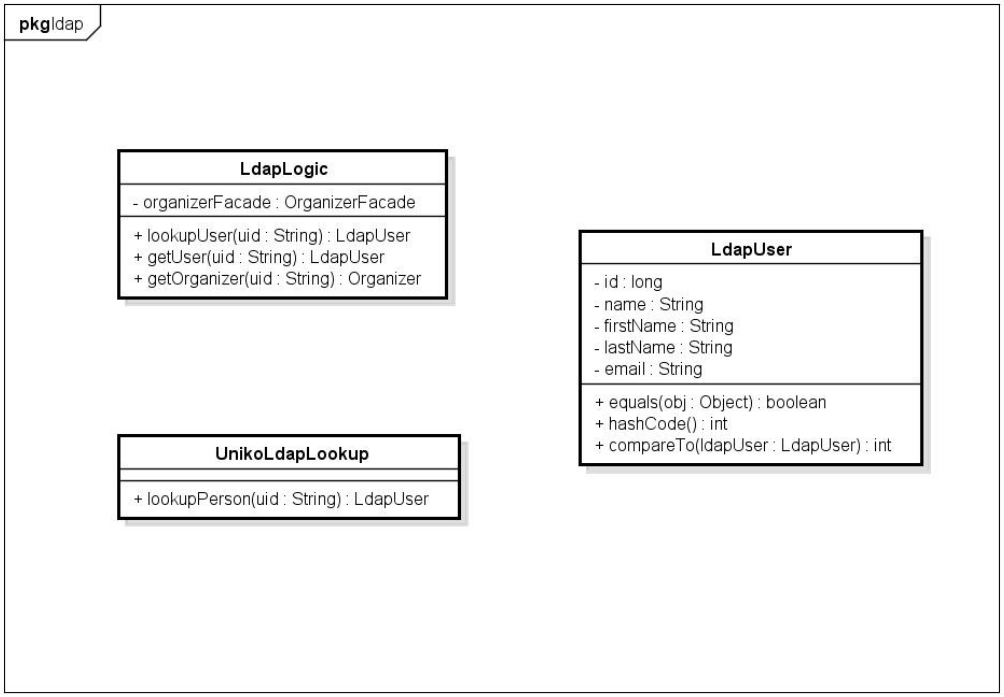


Components diagram not working

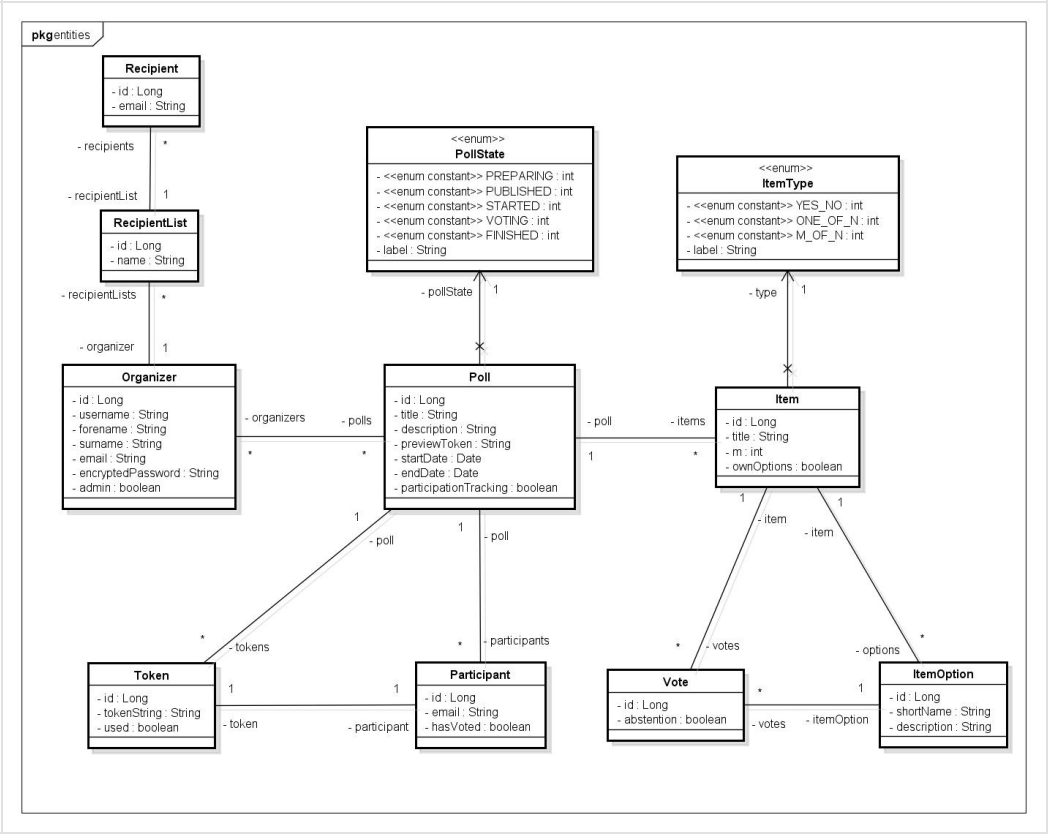
ZVotes-ejb



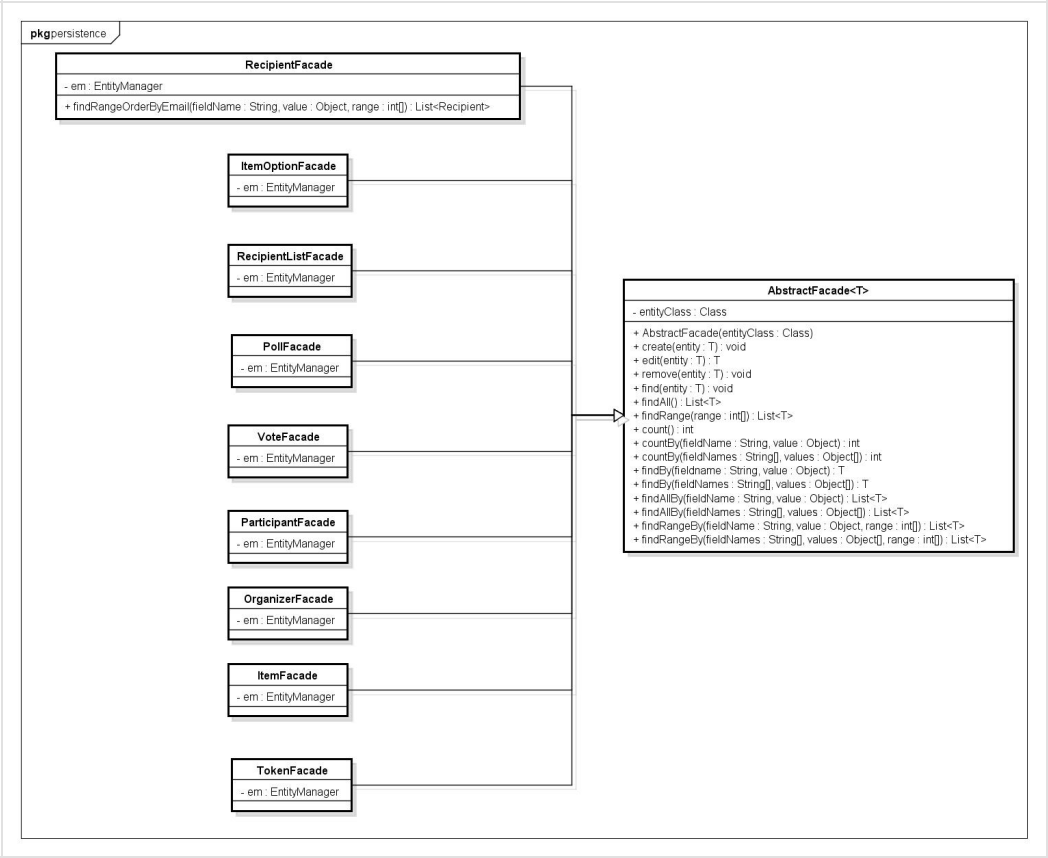
ejb pkg diagram not working



ldapstructure diagram not working

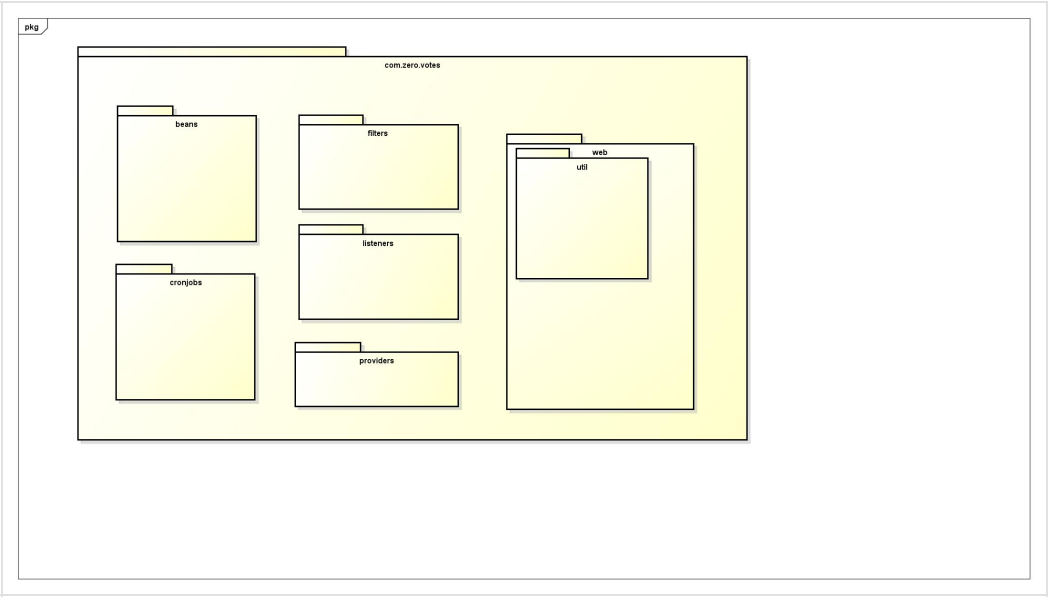


entities diagram not working

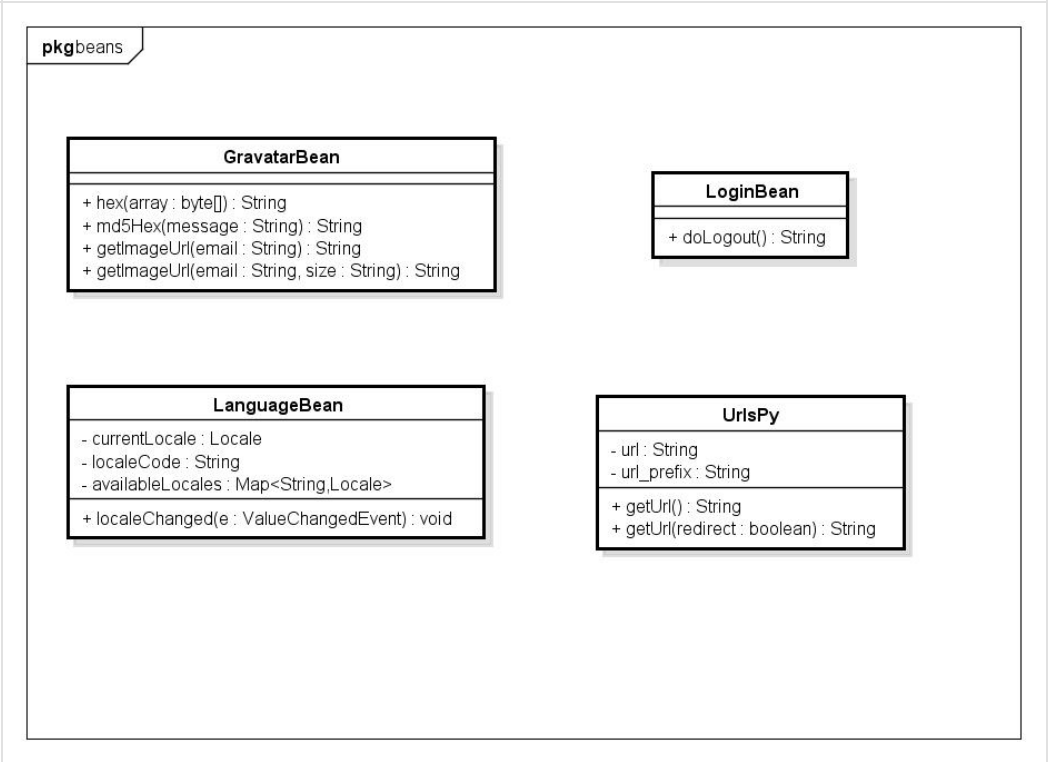


facades diagram not working

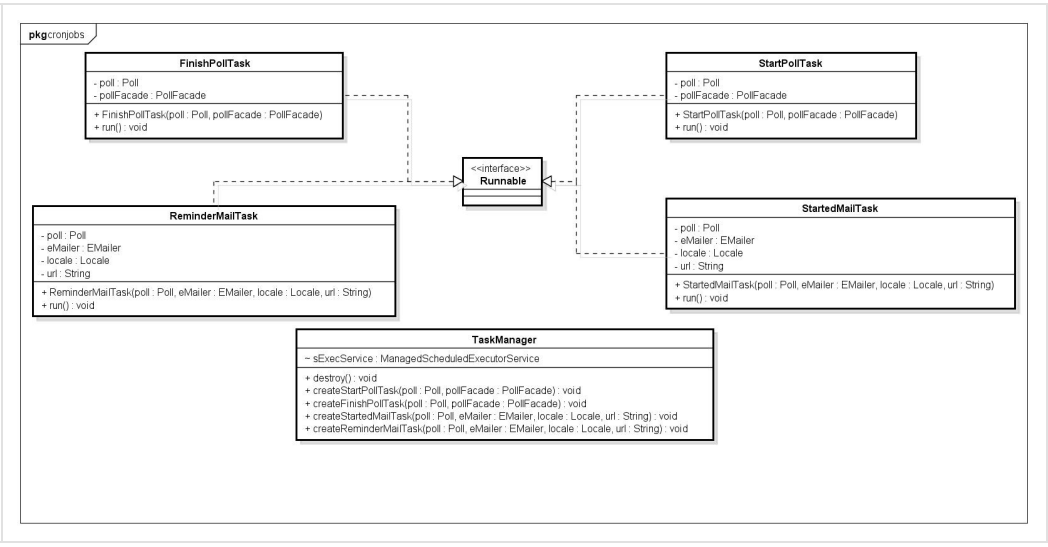
ZVotes-web



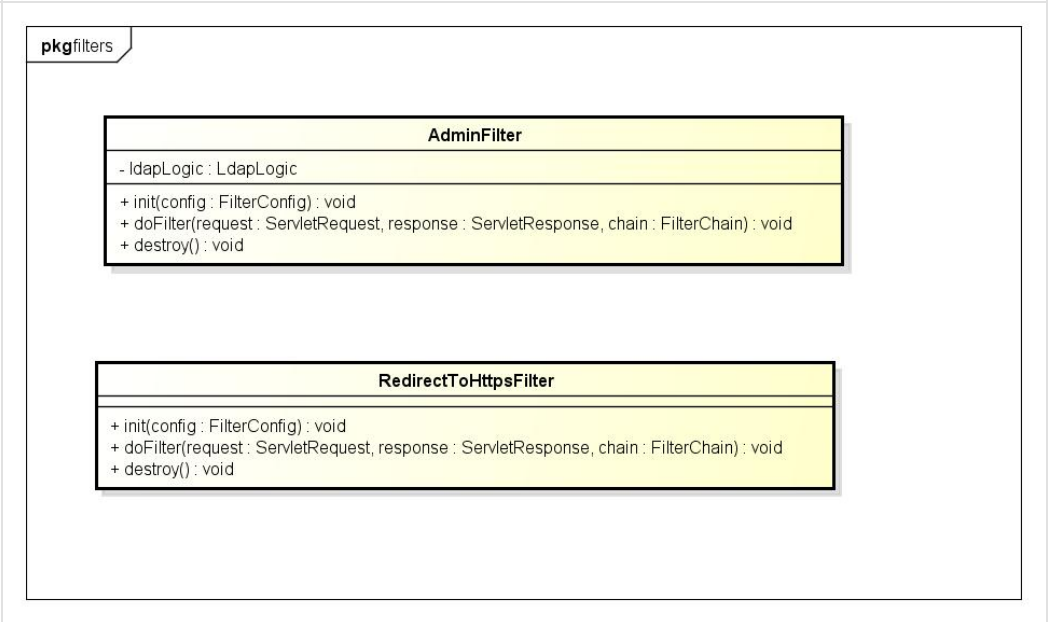
web pkg diagram not working



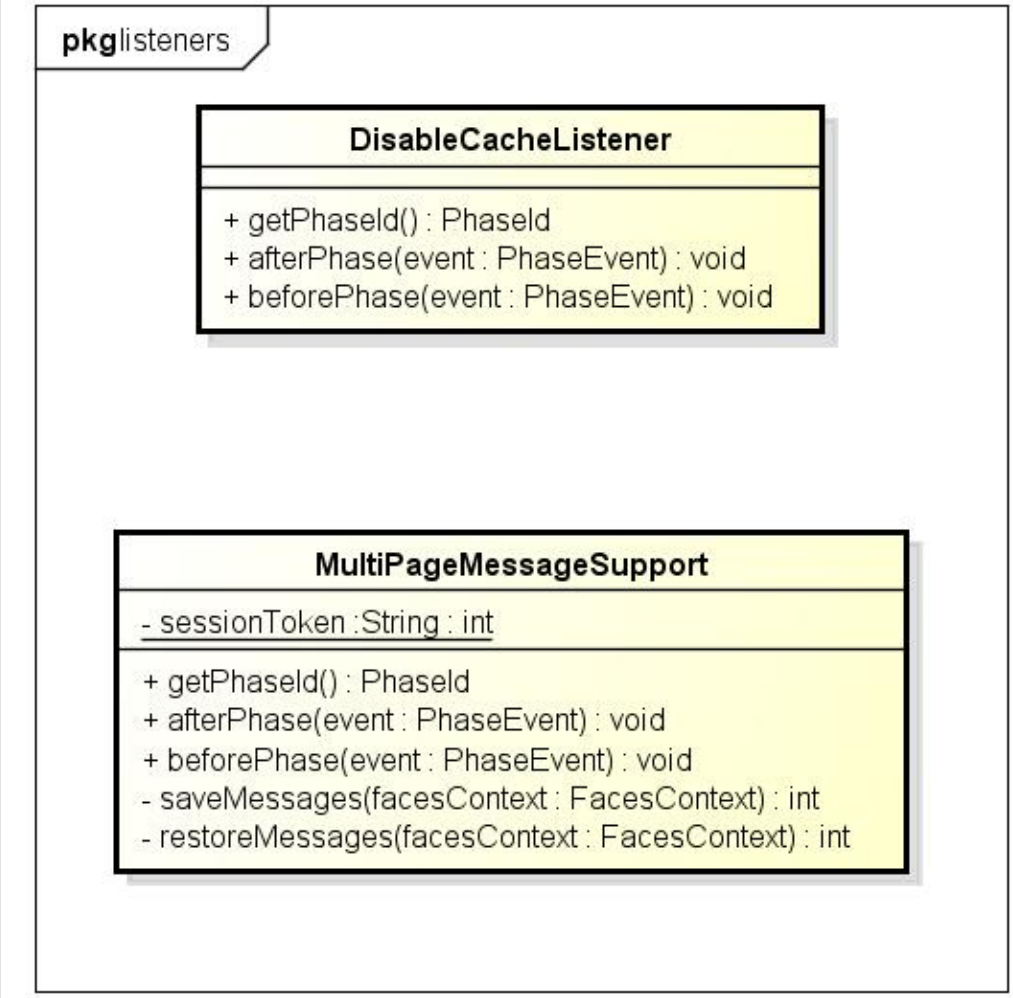
beans diagram not working



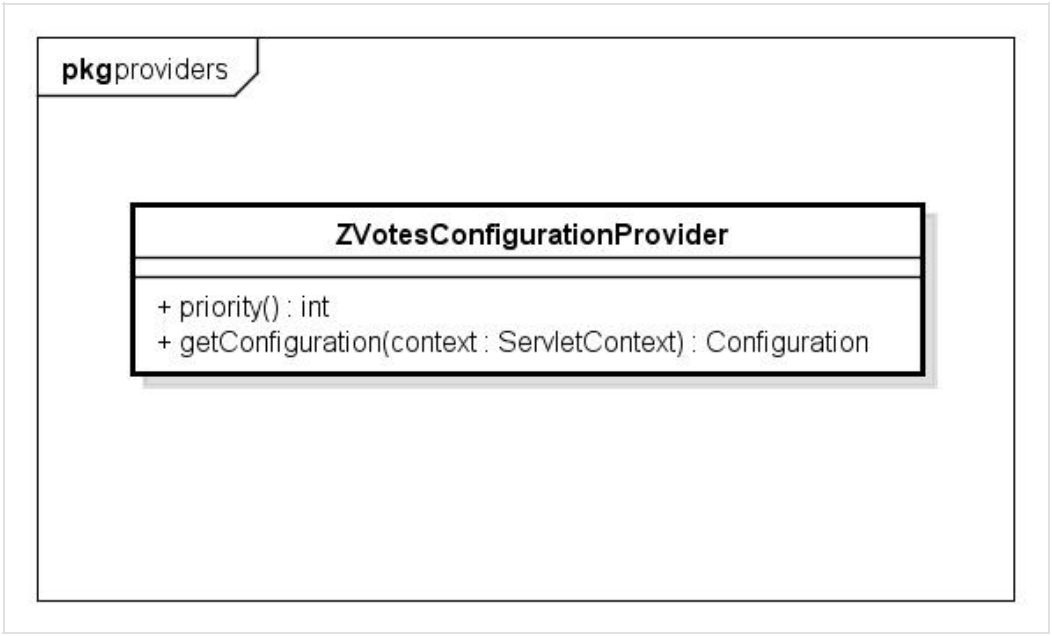
cronjobs diagram not working



filters diagram not working



listeners diagram not working



provider diagram not working