Requirements and Analysis Document

project: PocketPolitics

Requirements and Analysis Document for Democracy now

Version: 2.0

Date 2013-05-25

Authors: Group 13
Björn Bergqvist
Harry Eriksson
Johan Hasselqvist
Leif Schelin

This version overrides all previous versions.

Table of Contents

1 Introduction

In today's poltical climate, proxies i.e. politicians, make decisions on the voters behalf. However, it is hard for voters to get an idea of what decisions the politicians actually make and what effects these decisions have on them as well as others.

The problem springs from the fact that the sources that provide this information are usually biased. For example, in Sweden the magazine Aftonbladet which is a popular magazine with hundreds of thousands of readers have a strong left-wing bias which causes readers of this newspaper to adopt these ideas as well. One problem could be that Aftonbladet provides critical analysis of the right-wing parties, without ever writing about the success they make, and vice versa.

Ergo, we need a new way for voters to take part of an unbiased flow of political activity, quickly and with strong cohesion to today's immensely high-paced and high-volume information flow. Our solution for this is an internet portal that collects recent decisions from official information flows provided by the political instances that the users are interested in, and display this information in a user-friendly and pedagogic fashion.

1.1 Purpose of application

The immediate purpose is to provide an information channel for people who are interested in politics, but not politically involved. A channel that presents information about recent decisions and discussion from (for the user) relevant political instances, such as the parliament of the user's nation, and the the local political organs as well as global ones. All based on what the user is interested in.

The long-term purpose of this application is to make politicians as well as people in general, aware of the advantages that comes with involving citizens in political decisions. There is a strong inertia in today's political climate, where the politicians make guesses on what decision will make their voters happy (after all, most politicians are more interested in the public's opinion than practical progress - and this is something that we want to help change).

After the decisions have been made, the citizens are informed via different media's (as mentioned earlier, these are often biased and omit certain information and emphasize some other - because of economical and subjective reasons) after which they respond to the change with either positivity of negativity. So basically we have a flow that looks something like this:

Polticial decisions > The media's biased coverage > Voters make opinions and react

This means that the politicians do not know whether their decisions will be welcomed or hated

until long after the decisions have been made and economical resources has been spent on carrying these decisions out.

Our idea is that it would be in everyone's favour if a stronger cohesion about current discussions in the political instances would be introduced. A cohesion between decision-makers and the people who is affected, so any mistakes can be rectified *before* they're carried out - rather than after.

This will be to a larger extent than today be possible with the help of our application. The users (politically interested people, albeit a bit lazy - because of the fact that they do not manually visit all the diplomat-ridden hard-to-understand political websites) will be presented with political happenings in a simple way, and are then allowed to give their opinion on the topic through like and dislike buttons.

Because of this collection of public opinion, the politicians can easily take this into account when making decisions - instead of "shooting in the dark" as they do today.

To summarize, the purpose of the application is to help citizens understand what is going on in politics, as well as provide insight for decision makers into the public's opinion on different topics when making actual decisions to avoid making mistakes.

Furthermore, we hope that this will give everyone awareness of the benefits of crowd-sourced decisionmaking and in extent bring the world closer to a lobbyist-free direct democracy where power is distributed equally amongst all men and women - and biased decisionmaking is avoided to give the world's inhabitants bigger freedom and social equality.

1.2 General characteristics of application

The client-side of the application is a rather simple concept. The user will be greeted by a Reddit.com-like front page where different political happenings are presented. This data will be fetched from official political data-sources and remodeled into something a bit more presentable than what these sources currently make publically available.

Users can then interact with this information either by casting a vote on whether they agree, disagree or are uninterested in the political content of this information - or by making a comment on the topic (this will be a later implementation). Depending on how many people take interest (by voting, and commenting) in this topic it's presence will rise or fall, with the purpose of filtering out "unimportant"-topics that usually clutter political websites causing the users to quickly become uninterested.

In addition to this, the software will (based on previous voting characteristics of the user) carry out calculations on which party and politicians the user has the greatest agreement with - making the decision on whom to vote for easier for the user.

1.3 Scope of application

Initially, the scope will be limited to the following:

- Fetch data from the swedish Parliament (Riksdagen) and present this information as described in 1.2
    - Feed (Front page)
    - Detailed view of a specific topic
- User registration
- User opinion collection (Like/Dislike/Indifferent vote on each topic)
- Use the user statistics of votes to determine the agreement levels with different parties

If time allows us, we will try to implement as much of the following functionality as possible:

- Comment system
    - Filtering of comments (by similar methods to those that filter out irrelevant topics on the front page - like/dislike etc.)
    - Allowing users to report discriminatory comments
- Login with Facebook & Google accounts
- Smart auto-categorization of topics for use in a user-controlled filter-system and/or automatic filtering based on users interests
- In addition to recommending agreeable parties to vote for, also include functionality for recommending individual politicians
- Statistics about politicians and parties
- Fetching data from other instances than Riksdagen
- Alternative languages

1.4 Objectives and success criteria of the project

The app should use real authentic data and represent it correctly. The statistics should be reliable.

1.5 Definitions, acronyms and abbreviations

- Article, an event, decision or proposition discussed or voted upon in the Swedish Riksdag. Articles are displayed in a list and are likeable by the user.
- Comment, a short text written to comment an article or as a reply to a previous comment.
- Username, the name that will be displayed next to a comment referring to the author of the comment. Login will use email instead of username.
- PUL, Person-Uppgifts-Lagen. The Swedish law for individual privacy rights on the internet.

- Party, political party represented in the Swedish Riksdag.
- Politician, member of the Riksdag who has authority to vote and debate on Riksdag issues.
- GUI, Graphical User Interface.

2 Requirements

In this section we specify all requirements

2.1 Functional requirements

The user should be able to:

1. Create an account
2. Login
3. Mark an article with:
    a. Like
    b. Dislike
    c. Not interesting
4. View statistics and political concordance
5. Change account settings:
    a. Username
    b. Email
    c. Password
6. View app info

2.2 Non-functional requirements

2.2.1 Usability

A user should be able to read and like articles early in the development process.

The design should be intuitive for Android users and follow general and platform specific conventions for mobile devices and tablets.

2.2.2 Reliability

The app should work normally even though no connection with the server is established. Changed data should be sent automatically whenever a connection is established.

2.2.3 Performance

The app should always feel responsive even when loading data from the server. There must never be more than 1 sec response time from the GUI. Optimizing the loading procedure to run smoothly will be prioritized.

2.2.4 Supportability
The program should be flexible so it can easily be extended to a web page and iOS.

2.2.5 Implementation
The application should use standard Android programming principles supporting unrooted models from Android SDK 14.

2.2.6 Packaging and installation
The app should be available for download on Google Play.

2.2.7 Legal
The app should keep user data confidential and secure according to PUL.

2.3 Application models

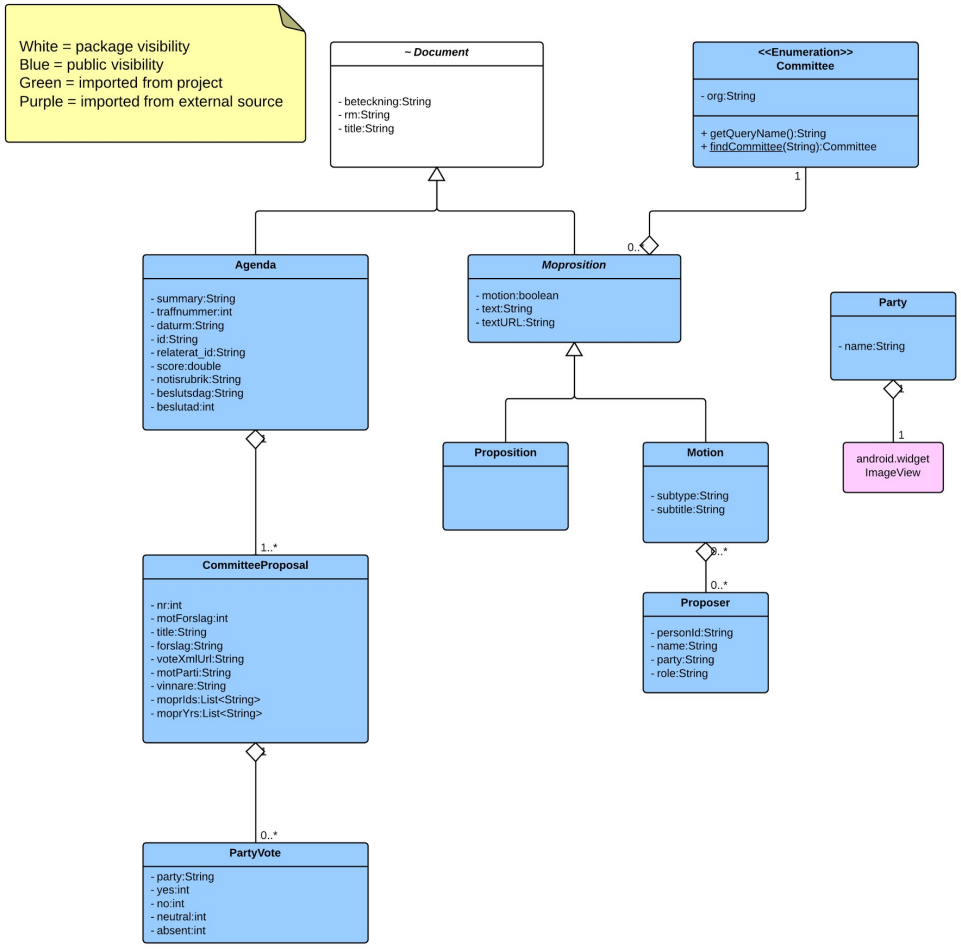
2.3.1 Use case model

// UML and a list of UC names (text for all in appendix)

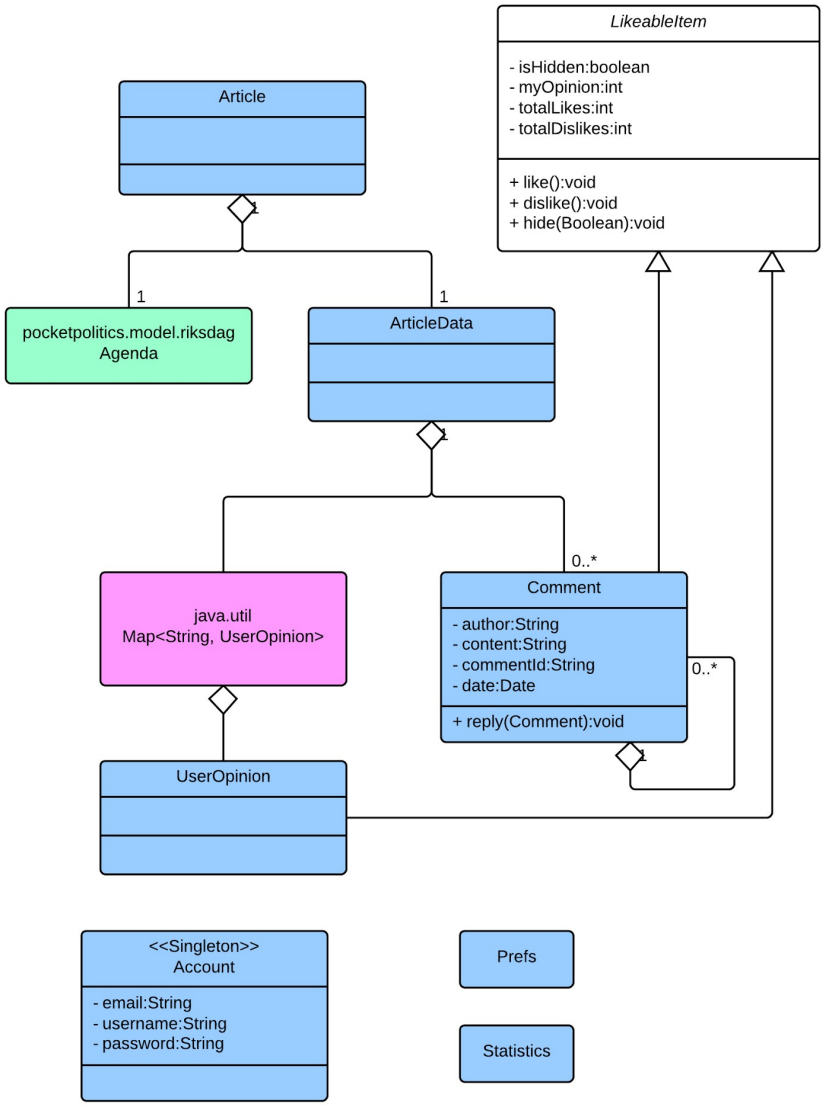
2.3.2 Use cases priority
Priority of use cases (highest most important):

1. Register and login
2. Like/dislike/mark article uninteresting
3. Statistical concordance
4. Change any account settings
5. Filter articles
6. View app info
7. Comments
    a. Comment article
    b. Reply to any comment
    c. Like/dislike comment
    d. Report comment
8. View party statistics
9. View recommended politicians
10. Login using Google or Facebook

2.3.3 Domain model

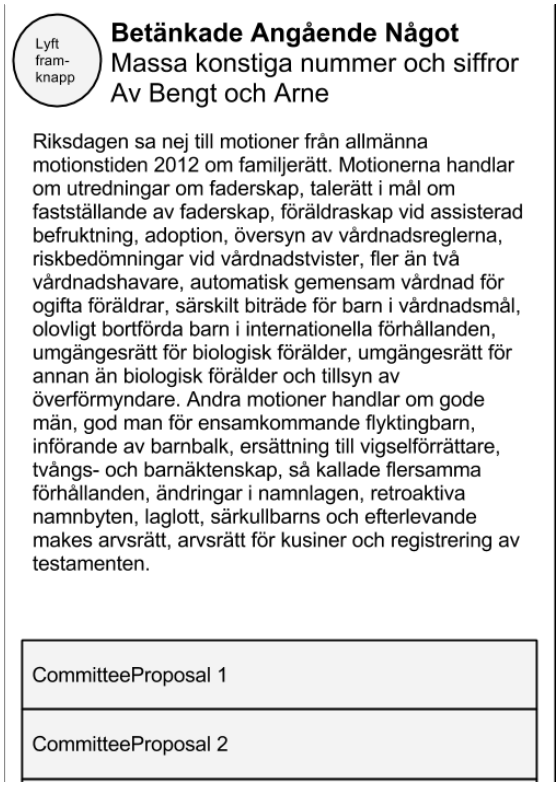


Model of the Riksdag classes (pocketpolitics.model.riksdag)

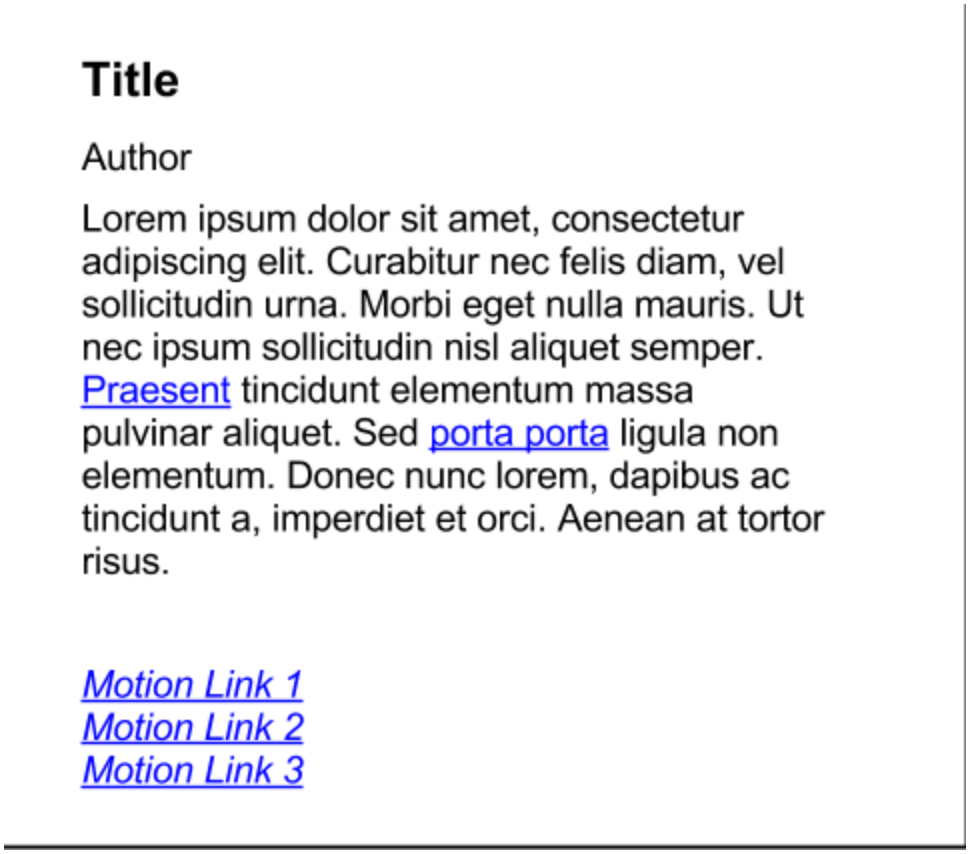


Class diagram of the user model (pocketpolitics.model.user)

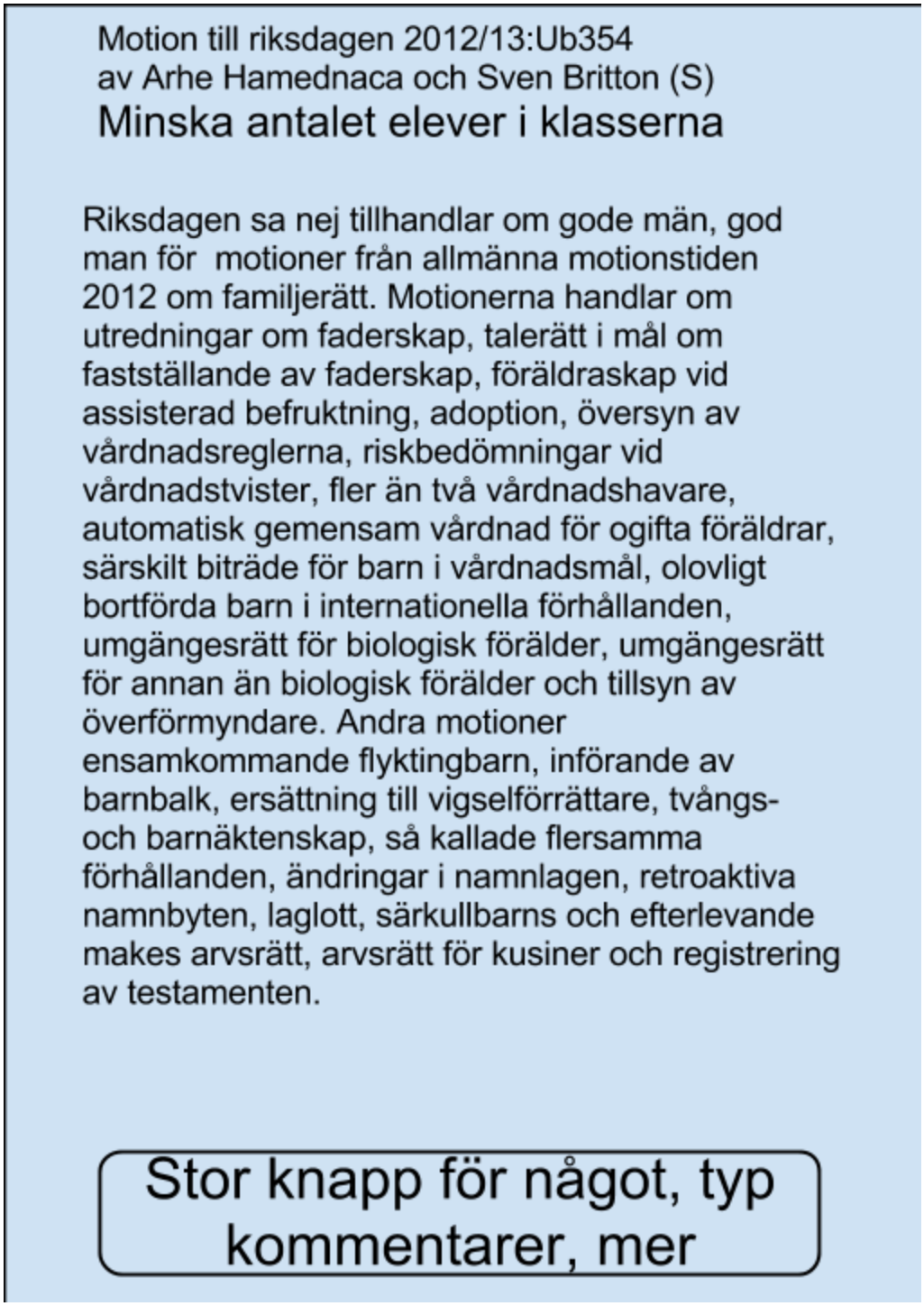
2.3.4 User interface



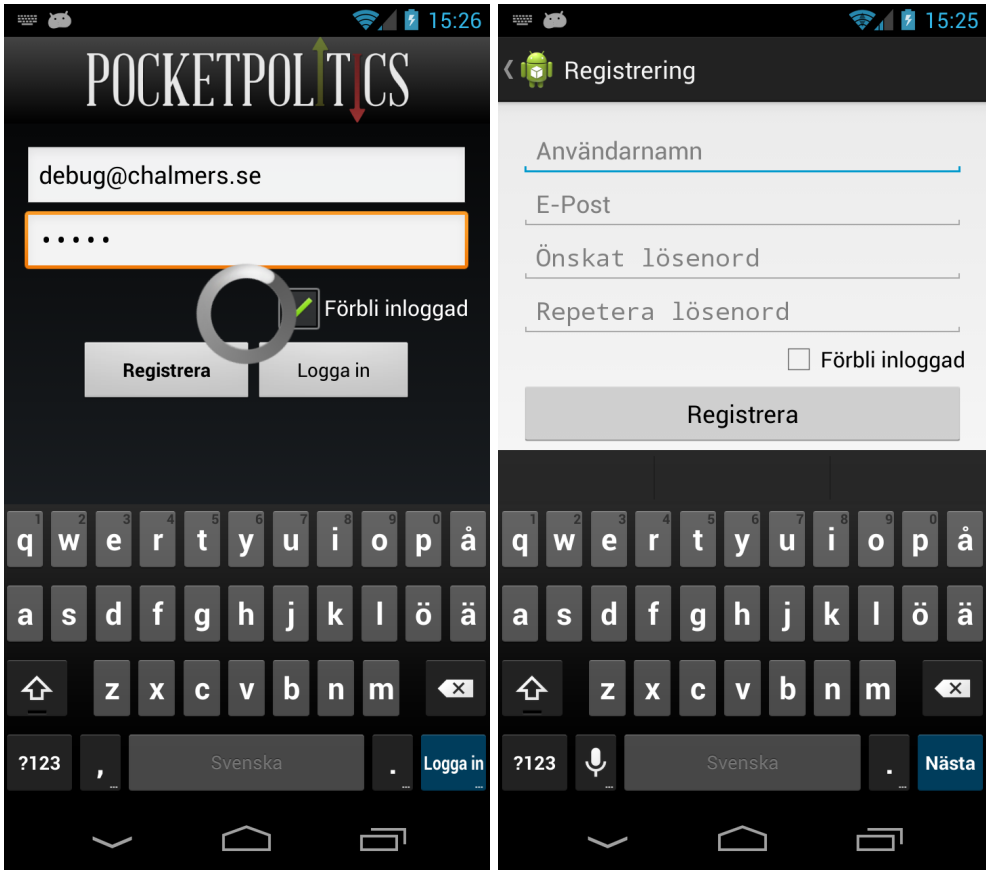
Sketch of how articles should be implemented into the GUI.



Sketch an expandable list item that displays committee proposals that should be implemented in a list inside the article view.



Sketch of how motions should be implemented into the GUI.



2.4 References

Appendix A: Use cases of high priority

1. Use case: Login

Summary: User is logging in. Assumes the user has an account already.
Priority: high
Extends: -
Includes: none
Participators: User

Normal flow of events

User has an account already and enters his data correctly.

#	Actor	System
1	Enter email.	
2	Enter password.	
3	Click login.	
4		Display progress indicator.
5		User is logged in. Display main view.

Alternate flows

Flow 1.1 User has selected to auto-login

#	Actor	System
1.1.1		Display progress indicator.
1.1.2		User is logged in. Display main view.

Flow 4.1 User wants to auto-login next time

#	Actor	System
4.1.1	Clicks “stay logged in”	

Exceptional flows

Flow 5.1 Wrong password or email during login.

#	Actor	System
5.1.1		Wrong password or email. Display message.

2. Use case: Register

Summary: User registers a new account.
Priority: high
Extends: -
Includes: none
Participators: User

Normal flow of events

User have no account, create new.

#	Actor	System
1	Click register.	
2	Enter email.	
3	Enter password.	
4	Enter password again.	
5	Click submit.	
6		Display progress indicator.
7		User is logged in. Display main view.

Exceptional flows

Flow 4.1 Password is too short.

#	Actor	System
4.1.1		Display message that the password needs at least 6 characters.

Flow 5.2 Entered passwords not equal.

#	Actor	System
5.2.1		Display message that the two submitted

		passwords are not equal. Please, retype.

Flow 5.3 Submitted email is no real email-adress.

#	Actor	System
5.3.1		Display message that the submitted email may be misspelled. Please, retype.

Flow 7.1 Email is already registered.

#	Actor	System
5.1.1		Display message that this account has already been created.

3. Use case: Logout

Summary: User is logging out.
Priority: high
Extends: -
Includes: none
Participators: User

Normal flow of events
User wants to log out.

#	Actor	System
1	Click settings button.	
2	Click logout.	
3		Display login-page.

4. Use case: Open article

Summary: Open article summary.
Priority: high
Extends: -
Includes: none
Participators: User

Normal flow of events

#	Actor	System
1	Clicks on an article in the list.	
2		Display progress indicator.
3		Display article summary and contents.

Exceptional flows

Flow 3.1 Operation failed

#	Actor	System
3.1.1		Display message that the summary cannot be retrieved due to technical problems.

5. Use case: Open motion/proposition

Summary: Open motion.
Priority: high
Extends: -
Includes: none
Participators: User

Normal flow of events

#	Actor	System
1	Clicks on a committee proposal.	
2		Displays text of the proposal and a list of buttons to referenced motions and propositions.
3	Click on a button displayed.	
4		Display progress indicator.
5		Displays complete text of the motion or proposition.

Exceptional flows

Flow 5.1 Operation failed

#	Actor	System
5.1.1		Display message that the summary cannot be retrieved due to technical problems.

6. Use case: Return up

Summary: Navigates back in the hierarchy.
Priority: high
Extends: -
Includes: none
Participators: User

Normal flow of events

#	Actor	System
1	Clicks Up-button or Back-button.	
2		Displays the previous view.

Exceptional flows

Flow 2.1 Already at top view

#	Actor	System
2.1.1		Close application.

7. Use case: Like a Motion/Proposition

Summary: User Likes a motion/proposition.
Priority: high
Extends: -
Includes: -
Participators: User

Normal flow of events

#	Actor	System
1	Click Like-button.	
2		Like-button becomes toggled. Text on button changes into “Liked”.

Alternate flows

#	Actor	System
1.1.1	Click Like-button.	
1.1.2		Like-button becomes toggled. Text on button changes into “Liked”. Dislike button becomes untoggled and its text changes to “Dislike”.

Exceptional flows
None

8. Use case: Dislike a Motion/Proposition

Summary: User Likes a motion/proposition.
Priority: high
Extends: -
Includes: -
Participators: User

Normal flow of events

#	Actor	System
1	Click Dislike-button.	
2		Dislike-button becomes toggled. Text on button changes into “disliked”.

Alternate flows

#	Actor	System

1.1.1	Click Dislike-button.	
1.1.2		Dislike-button becomes toggled. Text on button changes into “disliked”. Like button becomes untoggled and its text changes to “Like”.

Exceptional flows
None

Appendix B: Use cases of medium priority

9. Use case: Mark article as uninteresting

Summary: User marks article as uninteresting or irrelevant. This article will be hidden from view unless the user explicitly wants to see all articles.
Priority: medium
Extends: -
Includes: Enter article
Participators: User

Normal flow of events

#	Actor	System
1	Click “Not interesting”-button.	
2		This article disappears from view.

Alternate flows

Flow 1.1 First opens the detailed view of the article.

#	Actor	System
1.1.1	Click on article.	
1.1.2		Displays detailed text of the article.

Flow 2.1 Filter is set to “view all”

#	Actor	System

2.1.1		The article remains visible.

Exceptional flows
None

10. Use case: Add article filter

Summary: Filter and sort articles after time, most discussed and show articles marked “Not interesting”.
Priority: medium
Extends: -
Includes: Sorting and filtering.
Participators: User

Normal flow of events

#	Actor	System
1	Click the Filter-button.	
2		Display filter options such as sorting and filtering.
3	Edit the filter.	
4	Click OK.	
5		Display the filtered and sorted list of articles.

Alternate flows

Flow 3.1 Edit sorting option

#	Actor	System
3.1.1	Choose sorting options between “newest”, “oldest”, “most popular”, “most hated”, “most discussed”	

Flow 3.2 Edit filtering option

#	Actor	System

3.2.1	Choose filtering options between “last week”, “last month”, “last year”, “last mandate period”, “all”, “between dates...”	

Flow 3.3 Show hidden articles, previously marked as “Not interesting”

#	**Actor**	**System**
3.3.1	Click on “Show all articles”	

Flow 3.4 Hide articles, previously marked as “Not interesting”

#	**Actor**	**System**
3.4.1	Click on “Show only ‘interesting’ articles”	

Exceptional flows

Flow 3.2.1.1 Option “Between dates...” selected but the date is not recognized by the system

#	**Actor**	**System**
3.2.1.1.1		Display message: “Date not recognized, try using this format: yyyy-mm-dd”

11. Use case: Comment article

Summary: Write a comment to a specific article in the detailed article view. This can be used for discussion or giving feedback.
Priority: medium
Extends: -
Includes:
Participators: User

Normal flow of events

#	**Actor**	**System**
1	Click on an article.	
2		Display the article view.

3	Click in the comment field and write a comment.	
4	Press “submit”.	
5		Display the submitted comment alongside the other comments.

Alternate flows
None

Exceptional flows
None

12. Use case: Reply to comment

Summary: Reply to a comment made before. The target comment can either be a reply in itself or referring to the article.
Priority: medium
Extends: 9. Comment article
Includes:
Participators: User

Normal flow of events

#	**Actor**	**System**
1	Click Reply-button on a comment.	
2		Display a new comment field beneath the clicked comment.
3	Write comment and press Submit.	
4		Display the new comment alongside the other comments.

Alternate flows
None

Exceptional flows
None

13. Use case: Like/Dislike comment

Summary: Like or dislike comment. Liked comments will climb higher in the list of comments making them more visible while disliked comments will descend.
Priority: medium
Extends: 9. Comment article
Includes: Dislike comment.
Participators: User

Normal flow of events

Like comment

#	Actor	System
1	Click the Like-button on a comment.	
2		The like-button becomes green. The dislike-button becomes transparent gray.

Alternate flows

Flow 1.1 Dislike comment

#	Actor	System
1	Click the Dislike-button on a comment.	
2		The dislike-button becomes red. The like-button becomes transparent gray.

Exceptional flows

None

14. Use case: Change account settings

Summary: Change settings like email, password and username on comments
Priority: medium
Extends: -
Includes:
Participators: User

Normal flow of events

#	Actor	System
1	Click settings-button in navigation field.	
2		Display a list of all settings.
3	Click account-settings.	
4		Display current email, username and 3 textfields for changing password.
5	Edit information.	
6	Click OK.	
7		Display “settings saved”.

Alternate flows

Flow 5.1 Change password

#	Actor	System
5.1.1	Type the old password in the first field.	
5.1.2	Type the new password in the second field.	
5.1.3	Retype the new password in the third field.	

Exceptional flows

Flow 7.1 No connection to server

#	Actor	System
7.1.1		Display message: “The settings will be saved the next time you connect to the internet.”

15. Use case: View statistical concordance

Summary: Viewing concordance with political parties based on user statistics.
Priority: high

Extends: -
Includes:
Participators: User

Normal flow of events

#	Actor	System
1	Click statistics button in navigation field	
2		Display staple diagram of concordance for each political party.

Alternate flows
None

Exceptional flows
None

Appendix C: Use cases of low priority

16. Use case: View app info

Summary: View information about the app such as version.
Priority: low
Extends: -
Includes:
Participators: User

Normal flow of events

#	Actor	System
1	Click on AppInfo-button on navigation.	
2		Display dialog with app version, release date, developer group name and support email. Also display links to privacy policy statement, to user agreement, to this app on Google Play and to the Swedish Riksdag website.

Alternate flows
None

Exceptional flows
None

17. Use case: Report comment

Summary: Report comment violating user agreement such as offensive, vulgar or racistic.
Priority: low
Extends: 9. Comment article
Includes:
Participators: User

Normal flow of events

#	Actor	System
1	Click report-button on a comment.	
2		Displays message: “Comment by [user_xxx] is reported”.

Alternate flows
None

Exceptional flows
None

18. Use case: View party statistics

Summary: View detailed information and statistics about a political party in the Riksdag.
Priority: low
Extends: -
Includes:
Participators: User

Normal flow of events

#	Actor	System
1	Click on party-statistics in navigation.	
2		Display a list of all represented political parties in the Riksdag.
3	Click on a party.	
4		Display detailed information about the party such as political description, percentage of seats in the Riksdag, key members, vote attendance and vote unity. The concordance with the user as well as the voting concordance with other parties is displayed graphically with a staple diagram as well as numerically in percent.

Alternate flows
None

Exceptional flows
None

19. Use case: View recommended politicians

Summary: View recommendations for individual politicians who seem to be relevant for you.
Priority: low
Extends: -
Includes:
Participators: User

Normal flow of events

#	Actor	System
1	Click recommendation-button in navigation.	
2		Display list of politicians with name, accordance statistic and political party.

Alternate flows
None

Exceptional flows
None

20. Use case: Login using Google

Summary: Use a Google account to login.
Priority: low
Extends: 1. Login
Includes:
Participators: User

Normal flow of events

#	Actor	System
1	Click “login with google account”.	
2		Login using registered google account on device.

Alternate flows
See alternate flows for Use Case 1: Login.

Exceptional flows

Flow 2.1 Google account not registered on device.

2.1.1		Display fields for username and password.
2.1.1	Enter username and password. Press submit.	

See exceptional flows for Use Case 1: Login.

21. Use case: Login using Facebook

Summary: Use a Facebook account to login.
Priority: low

Extends: 1. Login
Includes:
Participators: User

Normal flow of events

#	Actor	System
1	Click “login with google account”.	
2		Login using registered facebook account on device.

Alternate flows
See alternate flows for Use Case 1: Login.

Exceptional flows

Flow 2.1 Facebook account not registered on device.

2.1.1		Display fields for username and password.
2.1.1	Enter username and password. Press submit.	

See exceptional flows for Use Case 1: Login.