

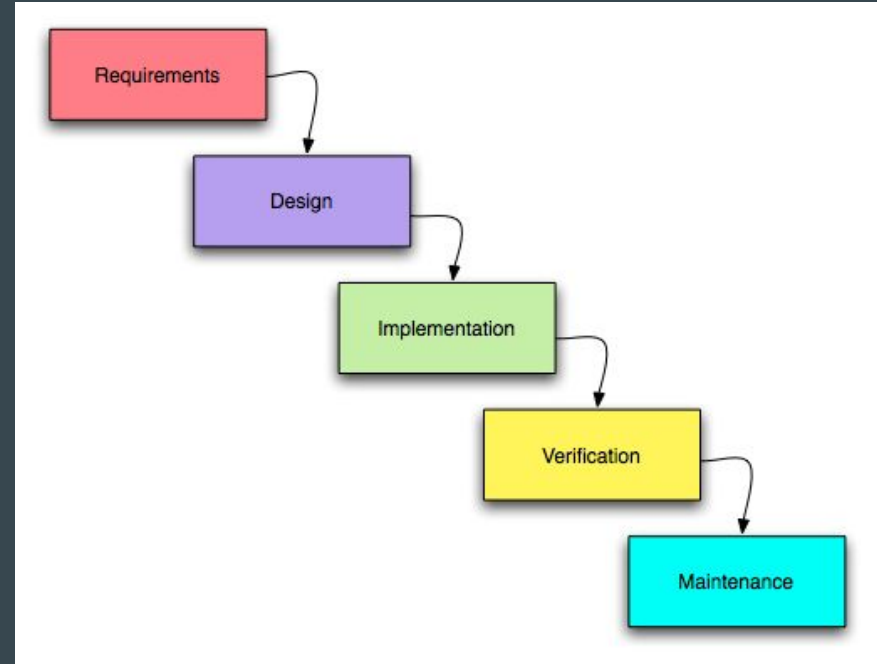
Group 1

...

Isabelle
Damian
Michael
Borna

Software Process

- Software Specification
- Software Design and Implementation
- Software Verification and Validation
- Software Evolution/Maintenance



Software Specification

- Understand the task at hand

Group Project Description

Senate Voting System

Disclaimer:

This is **not** a project on behalf of AEC, Australian Senate or Australian Government. The project description is only for an academic exercise. The public version of the laws are referred only as an example of the voting processes. Please **do not use any of the AEC, Australian Senate or Australian Government operational systems** for getting requirements or getting familiarity with the voting process.

Project Description:

The group project for this semester is to implement a web based voting system for a hypothetical body modelled on the Australian Senate, in which voters can either cast votes by voting for parties ("above the line") or for individual candidates ("below the line"). The system will allow an electoral commissioner to enter details of the candidates, including their order within their party grouping. It will allow individual voters to enter their votes either above the line, below the line but not both. It will correctly calculate the order in which senators should be elected. It will also allow the commissioner to manually exclude candidates if a recount should be ordered.

To understand the senate voting mechanism on which this project is modelled, it may be useful to refer to the following links about the actual voting rules for the Australian Senate:

The AEC published version of the senate voting rules (http://aec.gov.au/Voting/How_to_Vote/Voting_Senate.html).

candidate must have the following

- first name (given name)
- Last name (surname)
- ~~sex/gender~~ ^{party}
- associated ~~party~~ (e.g. Liberal)
- order by display below the line.
- state
- candidate ID
- (link to) candidate page.
- currently elected (Y/N)
- previously elected (Y/N)

100% necessary.

Software Specification

- Define the main functionality and constraints of software

~~→ voter~~
~~→ user wants to vote as non-admin user group 3.~~
327
→ user wants to log in as admin/~~user~~ voter.
→ as admin I want to add ~~the details~~
of a candidate.
→ as admin I want to remove a candidate.
→ as admin I want to edit the details of a candidate.
→ as an admin I want to view all candidates ~~user who has~~
~~amount of votes~~, filter list.
→ vote results page.
→ logout button.

Software Specification

- Discuss what is important to the prototype

vote system (view as voter)

→ 2 methods to vote ⇒ above the line
⇒ below the line.

→ above the line (6 choices)

→ below the line (12 choices)

→ a "submit" button

→ "clear" or "Refresh" button

→ "Back" or "Return" button for home page

→ error msgs. when user votes wrong.

→ display msg when user successfully voted.

→ list of parties & list of candidates in party. e.g.

→ text input or a method to input vote

above line	<input type="checkbox"/> name of party A	<input type="checkbox"/> party B	<input type="checkbox"/> party C	:	
below line	<input type="checkbox"/> candidate A1	<input type="checkbox"/> B1	<input type="checkbox"/> C1		
	<input type="checkbox"/> candidate A2	<input type="checkbox"/> B2	<input type="checkbox"/> C2	:	
	<input type="checkbox"/> candidate	<input type="checkbox"/> B3	<input type="checkbox"/> C3	:	
			<input type="checkbox"/> C4		

Software Specification

- Use “Group Project Iteration 1” as a guide

Learning Objectives

The objective is for each group member to practise the following software development process for some number of selected user stories:

1. allocate someone to the user story's issue log entry
2. make a new branch in which the user story will be implemented
3. on completion a pull request is lodged
4. another group member checks the user story implementation
5. if the implementation is acceptable the branch is merged and then deleted
6. the updated master branch is immediately published to Heroku

Software Design and Implementation

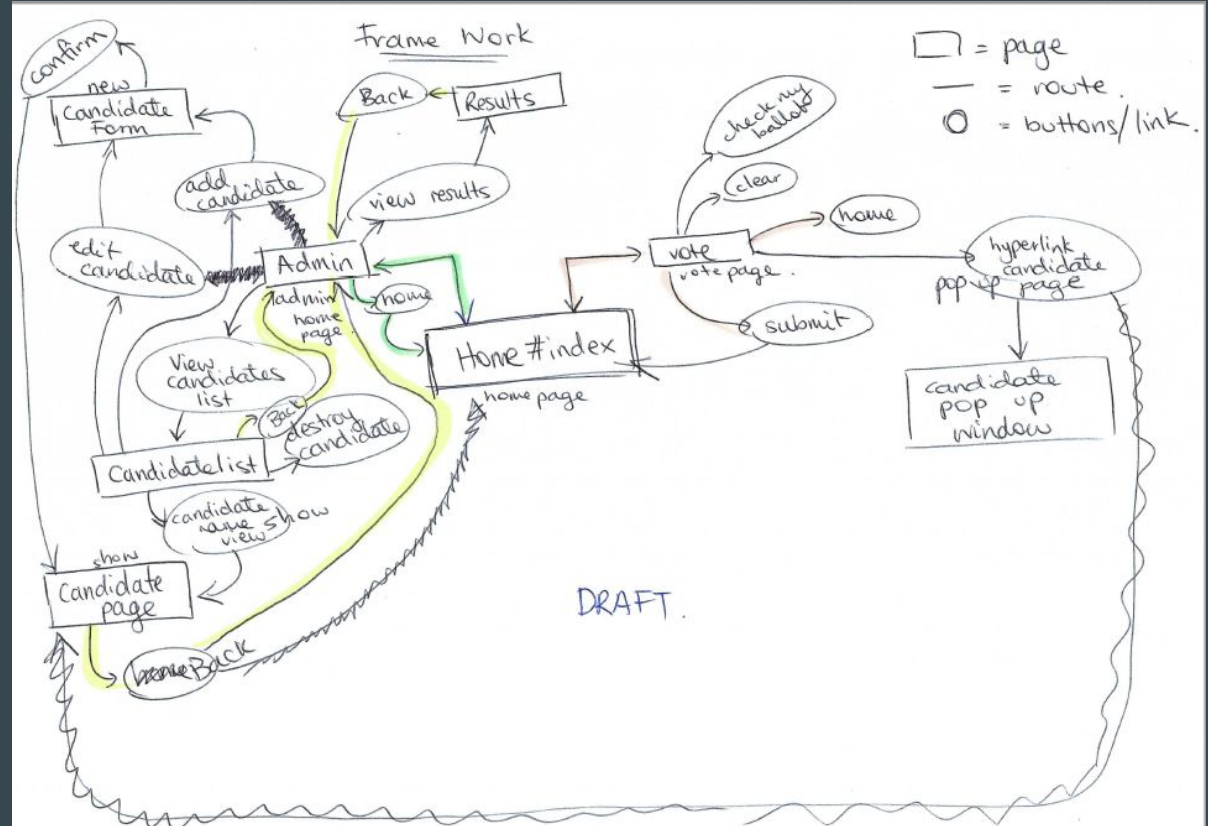
- Using user stories

The screenshot displays a Jira issue tracker interface. At the top, there is a search bar with the text "is:issue is:open" and a "Filters" dropdown. To the right, there are buttons for "Labels 9" and "Milestones 0", and a green "New issue" button. Below the search bar, a summary bar shows "15 Open" and "0 Closed" issues, along with columns for "Author", "Label", "Projects", "Milestones", "Assignee", and "Sort". The main list contains ten user stories, each with a checkbox, a green status icon, a title, and a timestamp. The stories are: "User Story: View Voter Ballots" (#16, 3 days ago), "User Story: Adding a Party" (#14, 17 days ago), "User Story: Voting System Choices" (#13, 20 days ago), "User Story: Voting Error Message" (#12, 20 days ago), "User Story: Voting Success Message" (#11, 20 days ago), "User Story: Viewing Candidate Page" (#10, 20 days ago), "User Story: Voting for Candidates" (#9, 20 days ago), "User Story: Voting for Party" (#8, 20 days ago), "User Story: View Results of Voting System" (#7, 20 days ago), and "User Story: Viewing Senate List" (#6, 20 days ago). Each story has a small icon to its right, representing a document or a person.

Issue Type	Status	Title	ID	Opened	By
<input type="checkbox"/>	Open	User Story: View Voter Ballots	#16	3 days ago	a1747931
<input type="checkbox"/>	Open	User Story: Adding a Party	#14	17 days ago	a1686874
<input type="checkbox"/>	Open	User Story: Voting System Choices	#13	20 days ago	a1686874
<input type="checkbox"/>	Open	User Story: Voting Error Message	#12	20 days ago	a1686874
<input type="checkbox"/>	Open	User Story: Voting Success Message	#11	20 days ago	a1686874
<input type="checkbox"/>	Open	User Story: Viewing Candidate Page	#10	20 days ago	a1686874
<input type="checkbox"/>	Open	User Story: Voting for Candidates	#9	20 days ago	a1686874
<input type="checkbox"/>	Open	User Story: Voting for Party	#8	20 days ago	a1686874
<input type="checkbox"/>	Open	User Story: View Results of Voting System	#7	20 days ago	a1686874
<input type="checkbox"/>	Open	User Story: Viewing Senate List	#6	20 days ago	a1686874

Software Design and Implementation

- First basic framework



Planning!

Homepage
Tuesday, 2 March 2021 12:54 PM

Logo

Vote Login as Admin

Home page link

Make Your Voice Heard

"Quote Here" - By Someone

Fun Image Here

Candidate Information Page

Friday, 21 May 2021 2:12 PM

"Name of Candidate" Page

Given Name

text here

Surname

text here

Party

text here

State

text here

text boxes

Back

Edit

button

Voting System

Page 1

To be elected as a senate/senator a candidate needs to gain a quota of the formal votes.

$$\text{quota} = \left\lfloor \frac{\text{number of formal ballot papers}}{\text{number of senators} + 1} \right\rfloor + 1$$

floor

Case Scenario:

If there exists 6 candidates & 3 senators to be elected & 100 ballot papers then,

$$\begin{aligned} \text{quota} &= \left\lfloor \frac{100}{3 + 1} \right\rfloor + 1 \\ &= \left\lfloor \frac{100}{4} \right\rfloor + 1 \\ &= 25 + 1 \\ \therefore &= 26 \end{aligned}$$

If one candidate achieves 30 votes in preference 1, then they get elected as they achieve more than the quota minimum.

Ballot paper (of 1 person)

Candidates:	A	B	C	D
	1	2	3	4

Current Total \Rightarrow 30 votes in total (A), 30 votes in total (B), 20 votes in total (C), 10 votes in total (D)

$$\text{surplus} = (\text{total 1st preference} - \text{quota})$$

$$\text{transfer_value} = \left(\frac{\text{surplus}}{\text{number of 1st preferences}} \right)$$

For the previous candidate with 30 votes

$$\text{surplus} = 30 - 26 = 4$$

$$\text{transfer_value} = \left(\frac{4}{30} \right) \approx 0.13$$

Their 2nd preference candidate B will receive the overflow of votes. However candidate C who has only 20 votes in total for their 1st preference that gets the overflow of votes.

Current Previous 1st Preference votes	A	B	C	D
	30	30	20	10

Current 1st Preference votes	A	B	C	D
	30	30.13	20	10

Now we calculate candidate B's overflow

for the ballot paper.

$$\text{surplus} = 30.1\bar{3} - 26 = 4.1\bar{3}$$

$$\text{transfer_value} = \frac{4.1\bar{3}}{30.1\bar{3}} \approx 0.1372$$

Now candidate C's 1st preference value is 20.1372

so from 1 ballot paper, we get:

Results

	A	B	C	D
Results	30	30.13	20.1372	10

For above the line, parties will be counted like candidates below the line.
Same thing different label.

For parties:

	A	B	C	D
Above the line	1	2	3	4
candidates:	1 2 3	4 5	6 7 8 9 10	11

The party preferences flow into below the line candidates & fill up like above.

Another party example:

	A	B	C	D
ABOVE THE LINE	1 Party A	2 Party B	3 Party C	4 Party D
BELOW THE LINE	9 name 10 name 11 name	3 name 4 name	5 name 6 name 7 name 8 name	1 name 2 name

Software Design and Implementation

- Creating a workflow

The screenshot displays a GitHub Projects workflow board. At the top, navigation tabs include Code, Issues (15), Pull requests, Projects (1), Wiki, and Insights. The 'Projects' tab is active, showing a workflow titled 'Workflow' updated 3 days ago. The workflow is organized into three columns: 'To Do' (4 items), 'In Progress' (4 items), and 'Completed' (5 items). Each item is a user story card with a title, ID, and author. A search bar 'Filter cards' is located above the 'Completed' column. To the right, a sidebar shows a search filter 'is:open' and search results for 'User Story: Voting Error Message' and 'User Story: Viewing Senate List'.

<> Code 1 Issues 15 Pull requests Projects 1 Wiki Insights

Workflow
Updated 3 days ago

4 To Do + ...

- 1 User Story: Viewing Candidate Page #10 opened by a1686874
- 1 User Story: Edit Candidate Information #5 opened by a1686874
- 1 User Story: Removing a Candidate #4 opened by a1686874
- 1 User Story: Voting Success Message #11 opened by a1686874

4 In Progress + ...

- 1 User Story: Voting System Choices #13 opened by a1686874
- 1 User Story: Voting for Candidates #9 opened by a1686874
- 1 User Story: View Results of Voting System #7 opened by a1686874
- 1 User Story: View Voter Ballots #16 opened by a1747931

5 Completed + ...

- 1 User Story: Logging in as Voter #2 opened by a1686874
- 1 User Story: Adding a Party #14 opened by a1686874
- 1 User Story: Adding a Candidate #3 opened by a1686874
- 1 User Story: Logging in as Admin #1 opened by a1686874
- 1 User Story: Voting for Party #8 opened by a1686874

+ Add column

Filter cards

+ Add cards

is:open

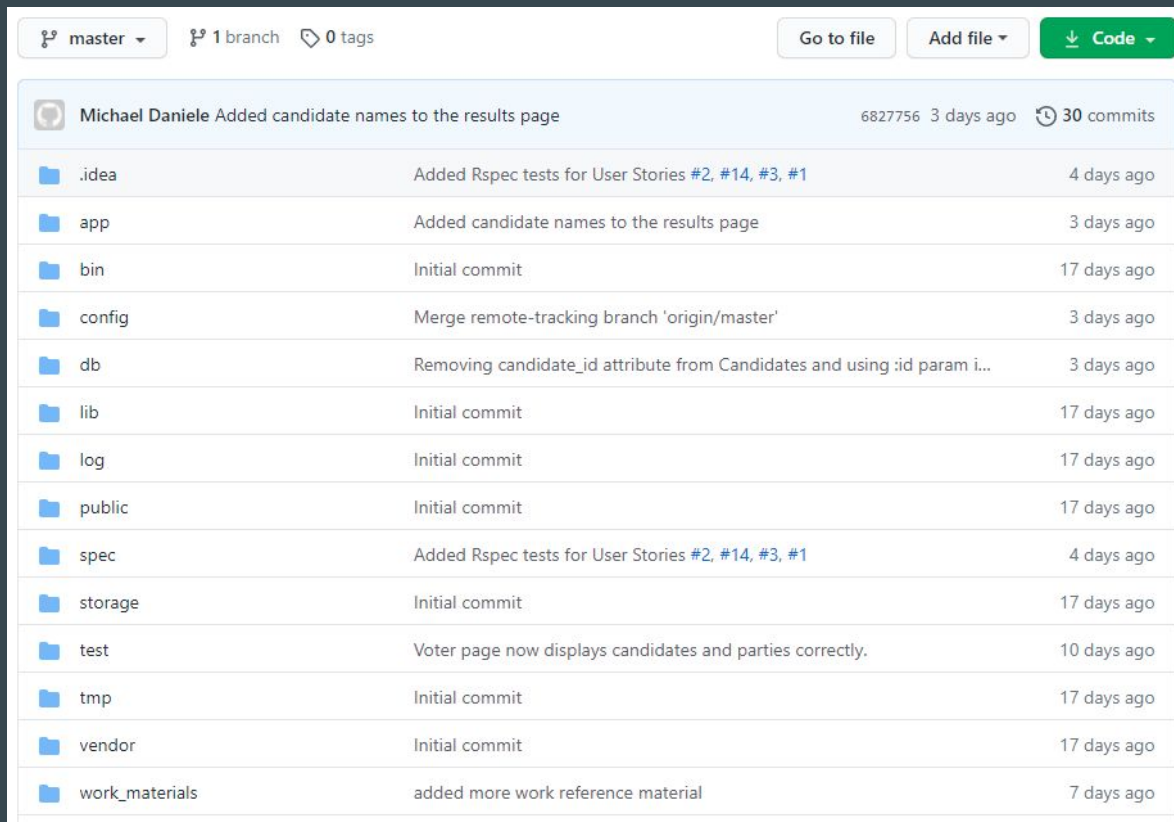
You can use the filters available in [issue search](#).

Search results

- 1 User Story: Voting Error Message #12 opened by a1686874
- 1 User Story: Viewing Senate List #6 opened by a1686874

Software Design and Implementation

- Initialise project on GitHub



The screenshot displays the GitHub interface for a repository. At the top, the 'master' branch is selected, with 1 branch and 0 tags. Buttons for 'Go to file', 'Add file', and 'Code' are visible. Below the repository name, a commit by Michael Daniele is highlighted, showing the message 'Added candidate names to the results page' and a commit hash of 6827756. The commit history table lists various files and their corresponding commit messages and timestamps.

File	Commit Message	Time Ago
.idea	Added Rspec tests for User Stories #2, #14, #3, #1	4 days ago
app	Added candidate names to the results page	3 days ago
bin	Initial commit	17 days ago
config	Merge remote-tracking branch 'origin/master'	3 days ago
db	Removing candidate_id attribute from Candidates and using :id param i...	3 days ago
lib	Initial commit	17 days ago
log	Initial commit	17 days ago
public	Initial commit	17 days ago
spec	Added Rspec tests for User Stories #2, #14, #3, #1	4 days ago
storage	Initial commit	17 days ago
test	Voter page now displays candidates and parties correctly.	10 days ago
tmp	Initial commit	17 days ago
vendor	Initial commit	17 days ago
work_materials	added more work reference material	7 days ago

Software Verification and Validation

- Written RSpecs

✓ 17 spec/requests/logins_spec.rb

```
...    @@ -0,0 +1,17 @@  
1  + require 'rails_helper'  
2  +  
3  + RSpec.describe "Logins", type: :request do  
4  +   describe "Logging in as a Voter" do  
5  +     it "Should show the voting page" do  
6  +       get voter_path  
7  +       expect(response).to have_http_status(200)  
8  +     end  
9  +   end  
10 +  
11 +   describe "Logging in as an Admin" do  
12 +     it "Should show the admin page" do  
13 +       get admin_path  
14 +       expect(response).to have_http_status(200)  
15 +     end  
16 +   end  
17 + end
```

Software Verification and Validation

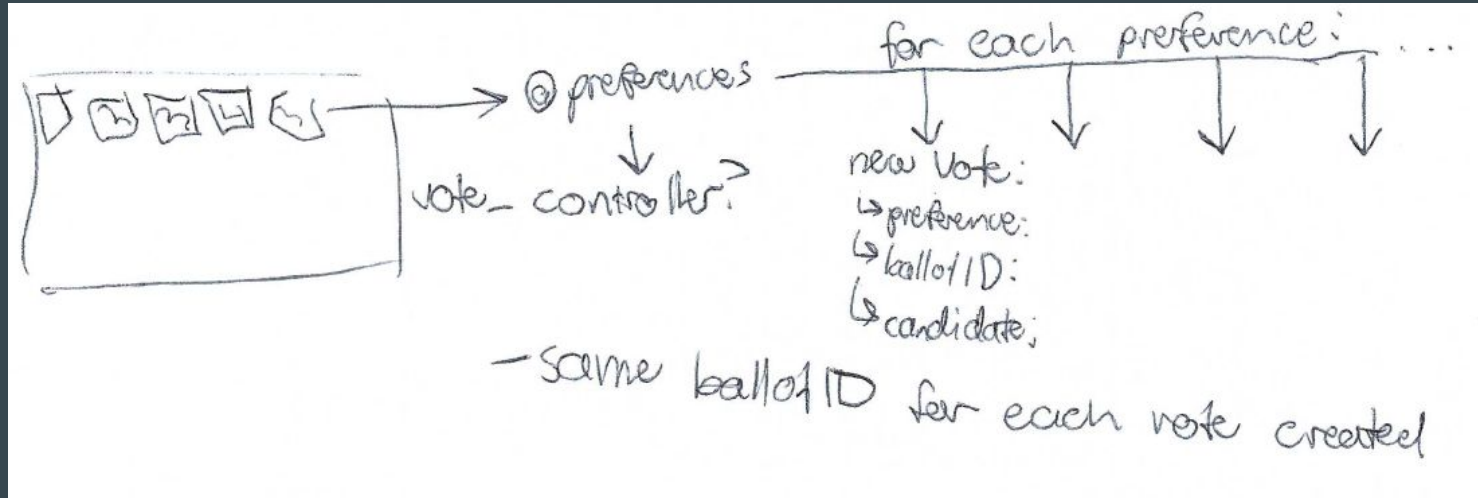
- Checking off workflow

The screenshot displays a GitHub Projects workflow board. At the top, navigation tabs include Code, Issues (15), Pull requests, Projects (1), Wiki, and Insights. The 'Projects' tab is active, showing a workflow titled 'Workflow' updated 3 days ago. The workflow is organized into three columns: 'To Do' (4 items), 'In Progress' (4 items), and 'Completed' (5 items). Each item is a user story with a status icon (green circle with an exclamation mark), a title, an ID, and the creator's username.

Column	Item	Status	ID	Opened by
To Do (4)	User Story: Viewing Candidate Page	Green circle with exclamation mark	#10	a1686874
	User Story: Edit Candidate Information	Green circle with exclamation mark	#5	a1686874
	User Story: Removing a Candidate	Green circle with exclamation mark	#4	a1686874
	User Story: Voting Success Message	Green circle with exclamation mark	#11	a1686874
In Progress (4)	User Story: Voting System Choices	Green circle with exclamation mark	#13	a1686874
	User Story: Voting for Candidates	Green circle with exclamation mark	#9	a1686874
	User Story: View Results of Voting System	Green circle with exclamation mark	#7	a1686874
	User Story: View Voter Ballots	Green circle with exclamation mark	#16	a1747931
Completed (5)	User Story: Logging in as Voter	Green circle with exclamation mark	#2	a1686874
	User Story: Adding a Party	Green circle with exclamation mark	#14	a1686874
	User Story: Adding a Candidate	Green circle with exclamation mark	#3	a1686874
	User Story: Logging in as Admin	Green circle with exclamation mark	#1	a1686874
	User Story: Voting for Party	Green circle with exclamation mark	#8	a1686874

Software Evolution/Maintenance

- It's just a prototype
- Future implementations



What did not work?

- GitHub forking, branching and pull requests
 - We all commit to master -> couple coding to minimise merge errors
 - Afeef was owner of the master so pull requests were a no go -> no branching out




git

What did not work?


- Semantic UI is not compatible with jQuery
- Semantic UI modal does not support dragging (proof below)


1 Answer

Active Oldest Votes


 Semantic UI does not support draggable modals at the moment like JQuery UI does.

1


 Default value of Detachable option is already true, which means that if you open a new modal the previous one will be moved to the background (= dimmer) and is not usable anymore (visualised with darker color).



Share Improve this answer Follow



answered Jun 12 '15 at 19:40

 **Marius Meissner**
26 • 2

What Worked?

- Workflow
- Allocation of work
- Agile approach
- Stand-up meetings



Workflow

[Code](#)[Issues](#) 15[Pull requests](#)[Projects](#) 1[Wiki](#)[Insights](#)

Workflow

Updated 3 days ago

4 To Do



- User Story: Viewing Candidate Page** ...
#10 opened by a1686874
- User Story: Edit Candidate Information** ...
#5 opened by a1686874
- User Story: Removing a Candidate** ...
#4 opened by a1686874
- User Story: Voting Success Message** ...
#11 opened by a1686874

4 In Progress



- User Story: Voting System Choices** ...
#13 opened by a1686874
- User Story: Voting for Candidates** ...
#9 opened by a1686874
- User Story: View Results of Voting System** ...
#7 opened by a1686874
- User Story: View Voter Ballots** ...
#16 opened by a1747931

5 Completed



- User Story: Logging in as Voter** ...
#2 opened by a1686874
- User Story: Adding a Party** ...
#14 opened by a1686874
- User Story: Adding a Candidate** ...
#3 opened by a1686874
- User Story: Logging in as Admin** ...
#1 opened by a1686874
- User Story: Voting for Party** ...
#8 opened by a1686874

Thank you for listening :)