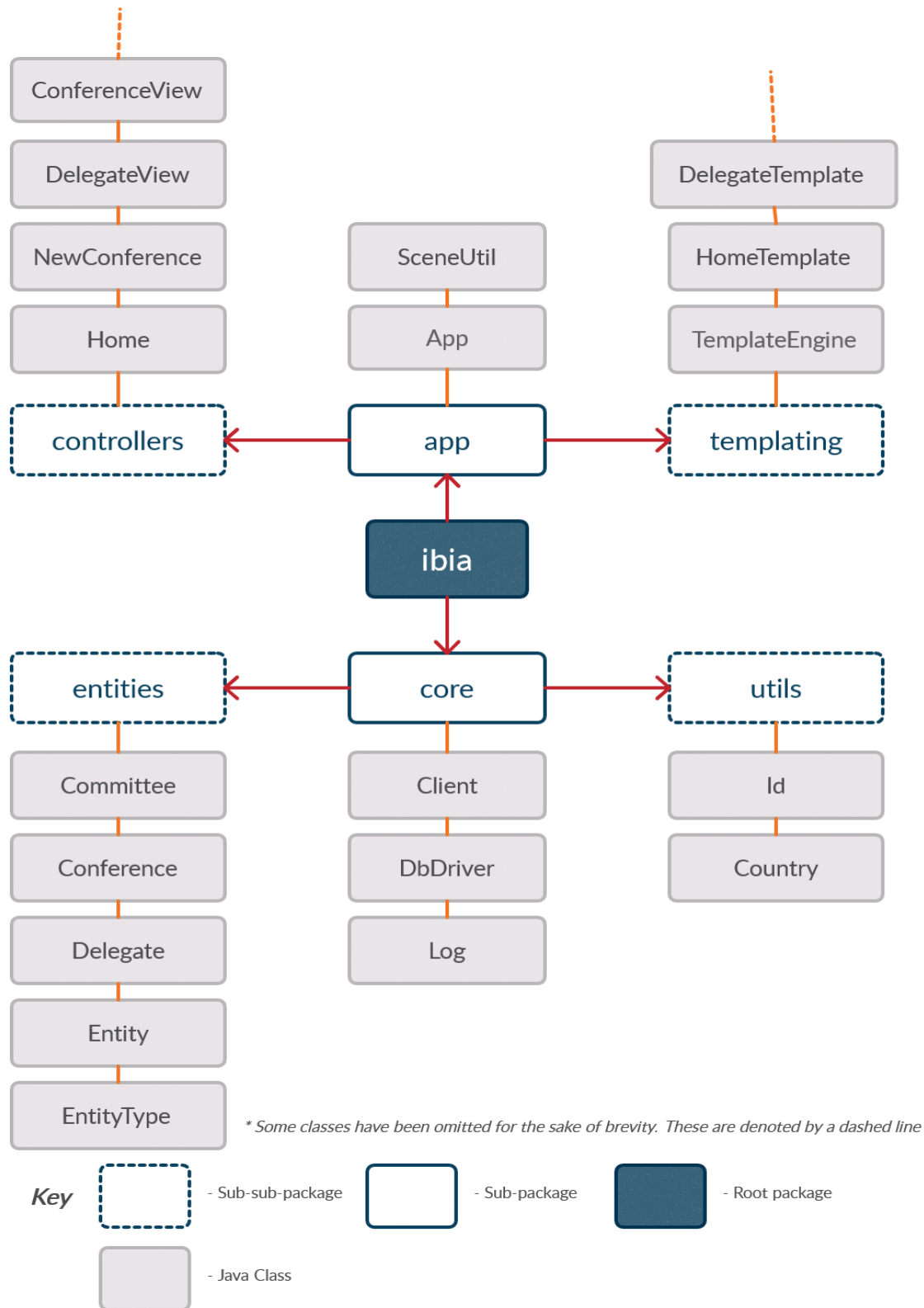


Criterion B – Design

Initial Package Structure Outline



ibia.app: Handles the JavaFX front-end, such as rendering views, handling JavaFX windows, and so on.

ibia.core: Handles the back-end, including database connections and data manipulation.

Database Normalization

Database tables were processed from an unnormalized form (UNF) to the third normal form (3NF) to ensure data integrity and reduce redundancy. The process has been illustrated below for the Committees table, with sample data:

UNF

Committee...				
id	Name	size	conferenceld	topics
COM12345	DISEC	15	CON12345	"Ending poverty", "Ensuring world peace", "Solving world hunger"

(extended below)

...Committee		
resolutions	resolutions_passed	main_submitters
"UK – Ending poverty", "India – Ensuring World Peace"	"DPRK – Solving World Hunger", "US – Ending Poverty"	DEL123, DEL456, DEL789, DEL012

1NF (First Normal Form): 1NF requires that each attribute contain only atomic values (no cell should contain multiple values). Multiple values can be separated by introducing new tables:

Committee			
id	name	size	conferenceld
COM12345	DISEC	15	CON12345

Topic		
id	committeeld	Topic
1	COM12345	"Ending poverty"
2	COM67890	"Ensuring world peace"
3	COM54321	"Solving world hunger"

Resolution				
id	main_submitter	committeeld	passed	topicId
1	DEL123	COM12345	TRUE	1
2	DEL456	COM54321	TRUE	3
3	DEL789	COM67890	FALSE	2
4	DEL012	COM12345	FALSE	1

2NF (Second Normal Form): In addition to 1NF, there should be no partial dependencies on concatenated keys (a primary key made of two columns). Tables with a single column acting as the primary key are already in 2NF.

3NF (Third Normal Form): In addition to 2NF, there should be no transitive dependencies on non-prime attributes, that is, all columns in a table should be functionally dependent on *only* the primary key. For

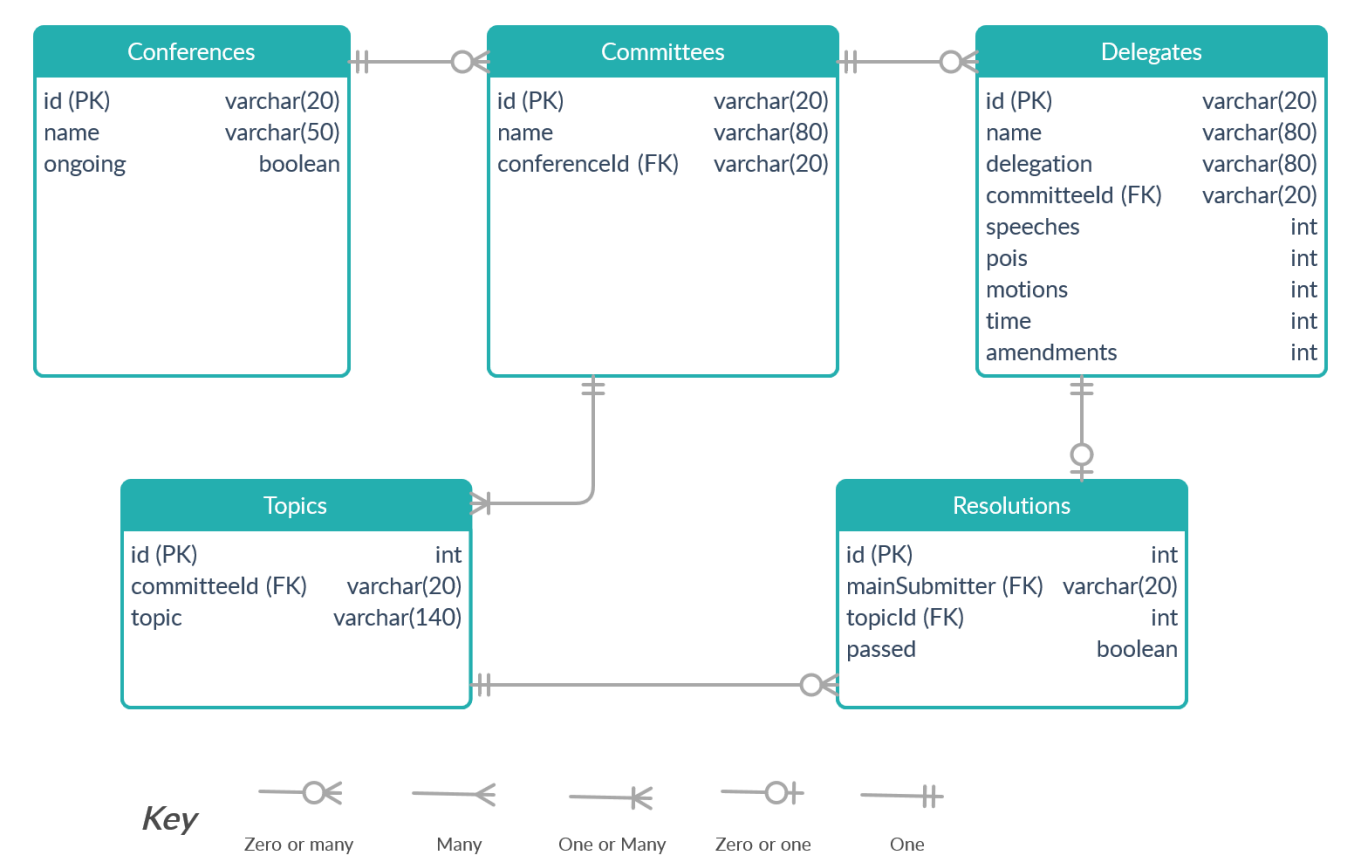
example, the committeeld column in the Resolutions table is redundant, as it can be obtained by looking at the topicId and obtaining the corresponding topic row. Removing such redundant columns gives:

Committee		
id	name	conferenceld
COM12345	DISEC	CON12345

Topic		
id	committeeld	topic
1	COM12345	"Ending poverty"
2	COM67890	"Ensuring world peace"
3	COM54321	"Solving world hunger"

Resolution			
id	main_submitter	passed	topicId
1	DEL123	TRUE	1
2	DEL456	TRUE	3
3	DEL789	FALSE	2
4	DEL012	FALSE	1

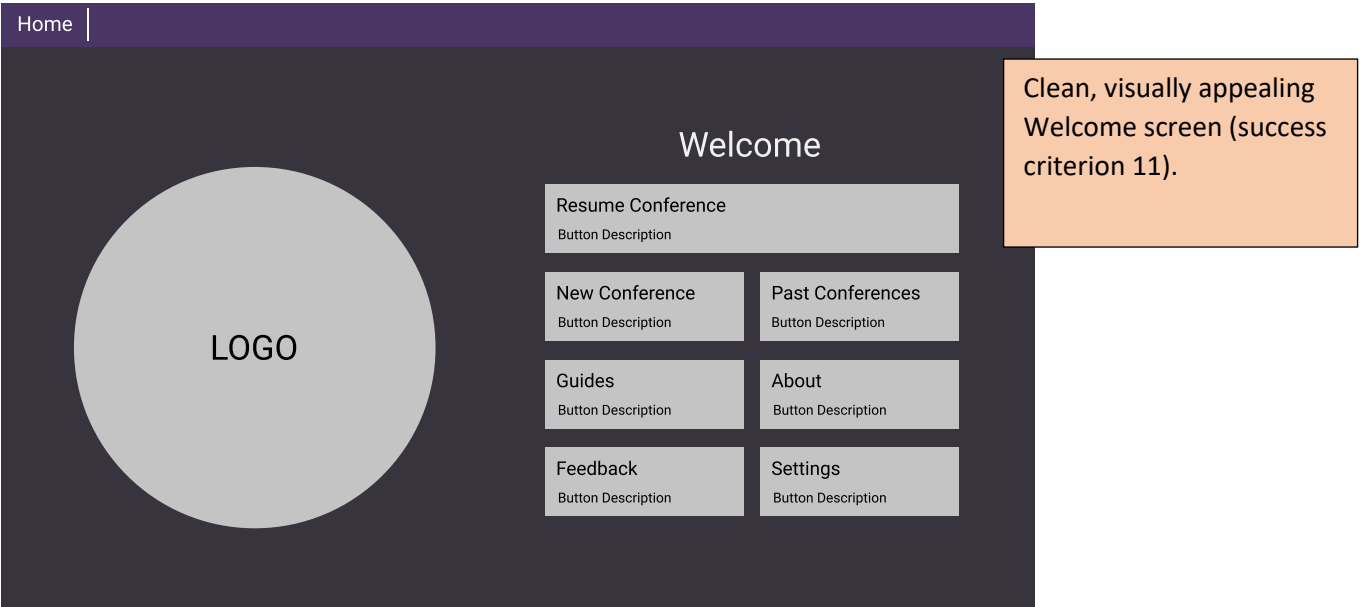
Database Entity Relationship Diagram (Crow’s foot notation)



User Interface Mockups

Mockups were made using Figma, a UI design & prototyping tool.
Note: These are from the initial design phase. The final product differs due to changes made later, based on client feedback.

Welcome Screen



Conference View

Home | ConferenceName

ConferenceName #conferenceId

EditDelete

Committees

Committee One Name

Committee Two Name

Committee Three Name

Add New Committee

Details

Opened: 01-01-2020

Status: Ongoing

Committees: 3

Delegates: 21

Finish Conference

Details for a specific conference, along with relevant controls.

Committee View

Home | ConferenceName > CommitteeName

CommitteeName #committeId

EditDelete

Delegates

	Speeches	Pols	Amendments	Motions
DelegationName1	13	6	2	3
DelegationName2	13	6	2	3
DelegationName3	13	6	2	3
DelegationName4	13	6	2	3

Add New Committee

Stats

Topics

Resolutions

Details for a committee, and relevant controls. To view statistics (success criterion 8), the “Stats” button opens a view similar to the Delegate View below, but with statistics for the chosen committee

Delegate View

Home | ConferenceName > CommitteeName > DelegationName

DelegationName#delegateld

These buttons are used to navigate around the application.

EditDelete

Details

Name:John Doe

Committee:DISEC

Speeches:13

Pols:6

Amendments:2

Motions:3

Resolutions Main-Submitted:0

Total Speaking Time0 minutes

Flag

This view shows details and statistics for a specific delegate (success criterion 8).

New Conference Popup

New Conference

Name

CancelCreate

Popup showing input fields for creating a new conference (The popups for creating new committees and delegates are similar, and therefore have been omitted)

Error Message Popup

Error

Error Message

OK

Displays important error messages to the user and improve user-friendliness.

Confirmation Message Popup

Confirm

Confirmation Message

CancelOK

Confirmation popup for irreversible or critical actions to prevent accidental actions and improve user-friendliness

Speech Timer

DelegationName speaking...

00:00

Start

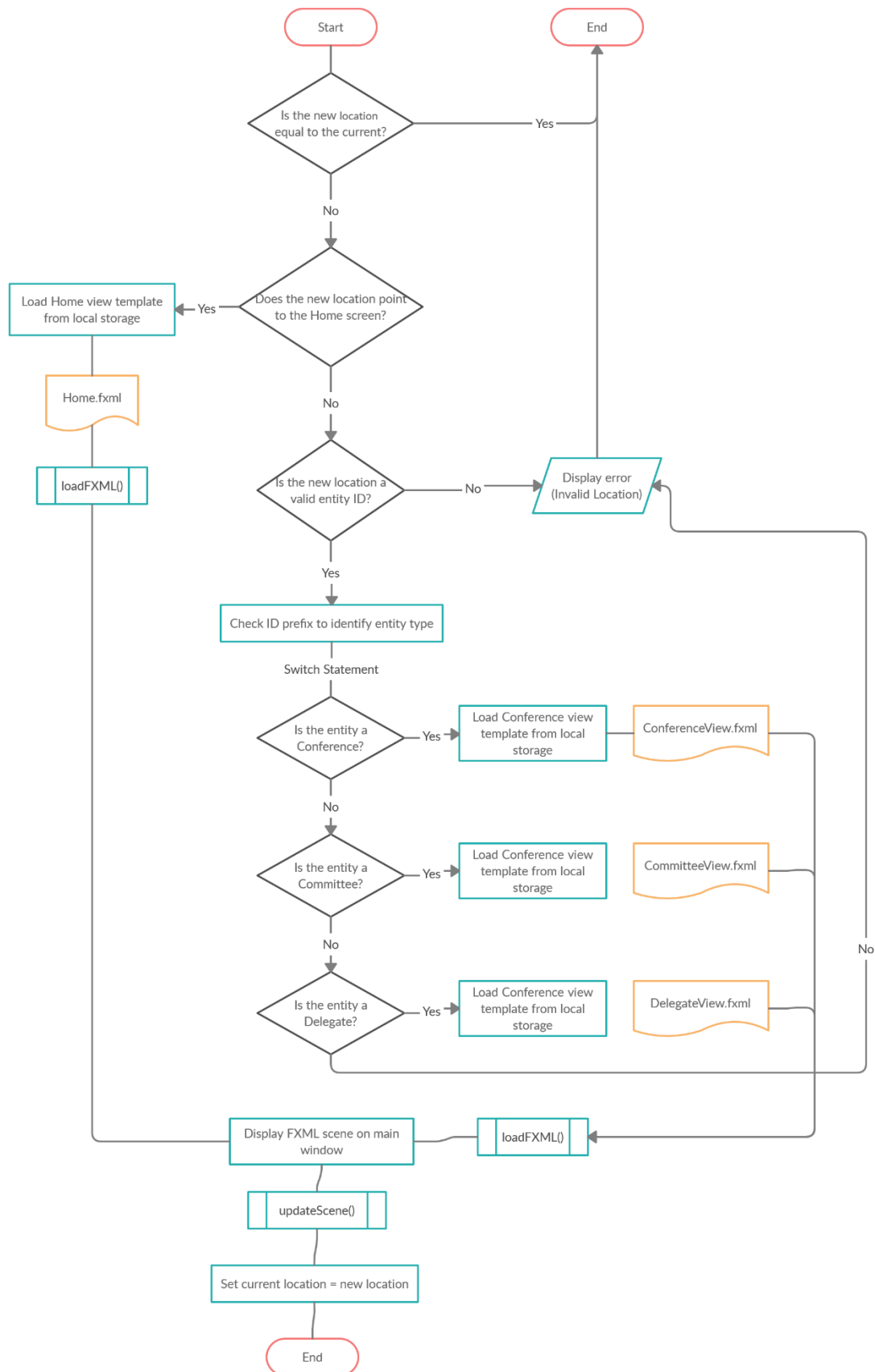
Stop

Finish

Popup for timing and controlling speeches (success criterion 6)

Flowcharts & Pseudo-code for Key Algorithms

The flowchart below shows the view navigation logic for the application, to allow smoothly changing the views displayed on the main window from any point in the application.



The following pseudo-code outlines a design pattern that will allow loading non-essential data in a fault-tolerant manner. This will be useful for multiple purposes, such as loading images and log data:

```
# fetch_data:   Performs some I/O operation
#               to obtain non-essential data.
#               It has a possibility of failing.
# process_raw:  Processes the raw data into a
#               a more useful form. It has a
#               possibility of failing as well.
# -----
# Set data to NULL to indicate that it has not
# been fetched or failed to fetch.
# It is important that this variable is outside
# the scope of the get_data function, otherwise
# it will be re-created and destroyed every time
# get_data is called. This would mean data would
# be fetched every time, leading to unnecessary
# I/O operations.
data = NULL

function get_data():
    # if check ensures data is only fetched
    # and processed if it is not already available.
    if data is NULL:
        try:
            raw = fetch_data()
            processed = process_raw(raw)

            # If data fetch + process succeeds,
            # store the result in the variable.
            # Since it is no longer null,
            # subsequent calls to this function
            # will return this same data without
            # delays.
            data = processed

        catch error:
            # If there was an error, output it
            # and set data back to null to indicate
            # the failure and try again next time.
            # A NULL-check may be necessary where
            # this function was originally called.
            output error
            data = NULL
    endif

    return data
end
```

Test Plan

#	Test Description	Example		Relevant Success Criterion
		Action	Expected Outcome	
1	The client should see a Welcome screen upon launching the product.	Press application icon / launch app.	The app opens to the Welcome screen, showing the logo and menu buttons ("New Conference", "Past Conferences" "Logs", "About", etc.)	11
2	The client should be able to create new conferences.	Press the "New Conference" button on the Welcome screen.	A popup shows asking for conference details. Once submitted, a new conference is created and shown on the main window.	1
3	The client should be able to edit conferences.	Press the "Edit" button on a Conference view	A popup shows asking for the new details. Once submitted, the conference is updated and shown on the main window.	1
4	The client should be able to delete conferences.	Press the "Delete" button on a Conference view.	A popup shows asking for confirmation. Upon confirmation, the conference and its committees and delegates are deleted, and the app goes back to the Welcome screen.	1
5	The client should be able to create new committees.	Press the "Add new committee" button on a Conference view.	A popup shows asking for committee details. Once submitted, a new committee is created and shown on the main window.	2
6	The client should be able to edit committee.	Press the "Edit" button on a Committee view.	A popup shows asking for new details. Once submitted, the committee is updated and shown on the main window.	2
7	The client should be able to delete committees.	Press the "Delete" button on a Committee view.	A popup shows, asking for confirmation. Upon confirmation, the committee and its delegates are deleted, and the app goes back to the Conference view.	2
8	The client should be able to create new delegates.	Press the "Add new delegate" button on a Committee view.	A popup shows, asking for delegate details. Once submitted, a new delegate is created and shown on the main window.	3
9	The client should be able to edit delegates.	Press the "Edit" button on a Delegate view.	A popup shows asking for new details. Once submitted, the delegate is updated and shown on the main window.	3

10	The client should be able to delete delegates.	Press the “Delete” button on a Delegate view.	A popup shows asking for confirmation. Upon confirmation, the delegate is deleted, and the app goes back to the Committee view.	3
11	The client should be able to create new topics.	Press the “New” button on the Topics view.	A new TextField shows which can be filled in with the new topic. Once filled in, it is saved to the database.	4
12	The client should be able to edit topics.	Press the topic TextField that needs to be edited.	Once the TextField for the topic is edited and shifted out of focus, the new topic is saved.	4
13	The client should be able to delete topics.	Press the topic TextField that needs to be deleted and clear all the text.	Once the TextField is empty and shifted out of focus, the topic will also be deleted from the database.	4
14	The client should be able to create new resolutions.	Press the “New” button on the Resolutions view.	Once the Delegate has been chosen from a dropdown menu, and the resolution title filled in, the new resolution is created, saved, and shown on the Resolutions view.	5
15	The client should be able to edit resolutions.	Press the resolution details that need to be edited.	Once the new details are filled in, and are shifted out of focus, they will be saved to the database.	5
16	The client should be able to delete resolutions.	Press the “Delete” button next to a resolution.	The corresponding resolution is removed from view and deleted from the database.	5
17	The client should be able to start and stop a timer for speeches.	Press the “Timer” button on the Committee view.	A popup shows, with the timer set to 0:0, with a start/stop button for controlling the timer.	6
18	The client should be able to update the number of different actions for each delegate.	Press the TextField for the delegate in the relevant action column.	The TextField can be updated with the new number of actions, which are saved once the TextField is shifted out of focus.	7
19	The client should be able to see basic statistics for all entities.	Open the relevant entity’s page on the main window.	The main window for each entity displays basic statistics for the entity (e.g number of committees in a conference, list of delegates in a committee, total actions for a delegate, etc.)	8

20	The client should be able to mark conferences as finished or ongoing.	Press the "Finish/Re-Open Conference" button on the Conference view.	Upon pressing the button, the status of the Conference is updated to Finished or Ongoing.	9
21	The client should be able to access finished conferences and delete them when needed.	Press the "Past Conferences" button on the Welcome screen.	A popup shows with the list of conferences marked "Finished". These can be accessed by clicking on them, and then deleted like any other conference.	10
22	Form validation when creating a new conference, committee, or delegate.	Press "Submit" on the popup for creating a new conference, committee, or delegate.	Before the form action is performed, the data is validated by the form controller to check for errors (see test plans 23, 24).	13
23	Show error if invalid data is submitted when creating a new conference, committee, or delegate.	Press "Submit" when the entity name is empty, or above the character limit.	An error popup is shown, informing the client that fields cannot be empty, or longer than the limit.	13
24	Handle error if invalid data is entered into the delegate actions field on a Committee view.	Enter non-numeric text into the field and shift it out of focus, such as "abc".	Data shown on the field is reverted to the last valid version saved to the database.	13
25	The client should be able to easily navigate around the app.	Perform any action that updates the main window to a new page.	The navigation bar located at the top is always up-to-date and can be used any time to easily navigate to pages relevant to the current view.	12