COSC 499 Capstone Project Report

AGMeeting(1)

Submitted by:

Carson Ricca

Ryan Keilty

Chester Ng

Garry Feng

Under the Guidance of

Dr. Gema Rodríguez-Pérez

The University of British Columbia Okanagan

October 18, 2021

Table of Contents

High-level Project Description

AGMeeting is a meeting software designed to streamline the process of company meetings in accordance with Robert's Rules. The software is designed to be used on any device with internet browsing capabilities. The users of the software are separated into 3 categories, Participant, Moderator, and Administrator. The software's main function is to facilitate efficient communication during meetings through the use of an interactive agenda. The moderator sets up the meeting structure by adding items to the agenda and then the participants use this agenda to vote or discuss various items or proposals. The target user group for this software would be mid to large scale companies that have meetings frequently, as well as companies that have employees traveling that can participate remotely. This software can also be used by smaller companies or startups through the light version of the application which removes some of the larger scale features. This version of the application requires the Super Administrator, Barry, to be set up.

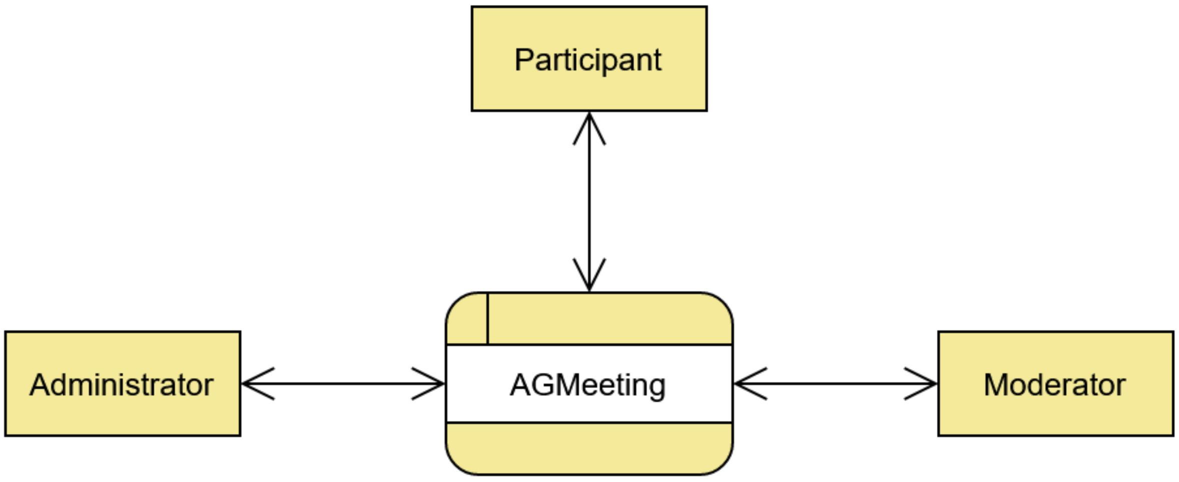
Target User Groups

Mid to Large Scale Companies: This user group will use the software in its full capacity. This is the focus user group for the program as many of the features cater to large scale meetings with many participants. This user group will most likely be using features such as templates while fully making use of all three roles. This user group is also more likely to have remote participants as larger sized companies may have employees working in different regions.

Smaller Companies and Startups: This user group will more likely benefit from the stripped down light version of the application as their meetings will more likely consist of less agenda items and have fewer members. This user group is also more likely to have moderators and administrators acting as participants or having a single person function as both the administrator and moderator. This user group is also more likely to use the application for different meeting types other than AGMs. For example, anything from weekly stand up meetings to brainstorming sessions could be conducted through the AGMeeting software.

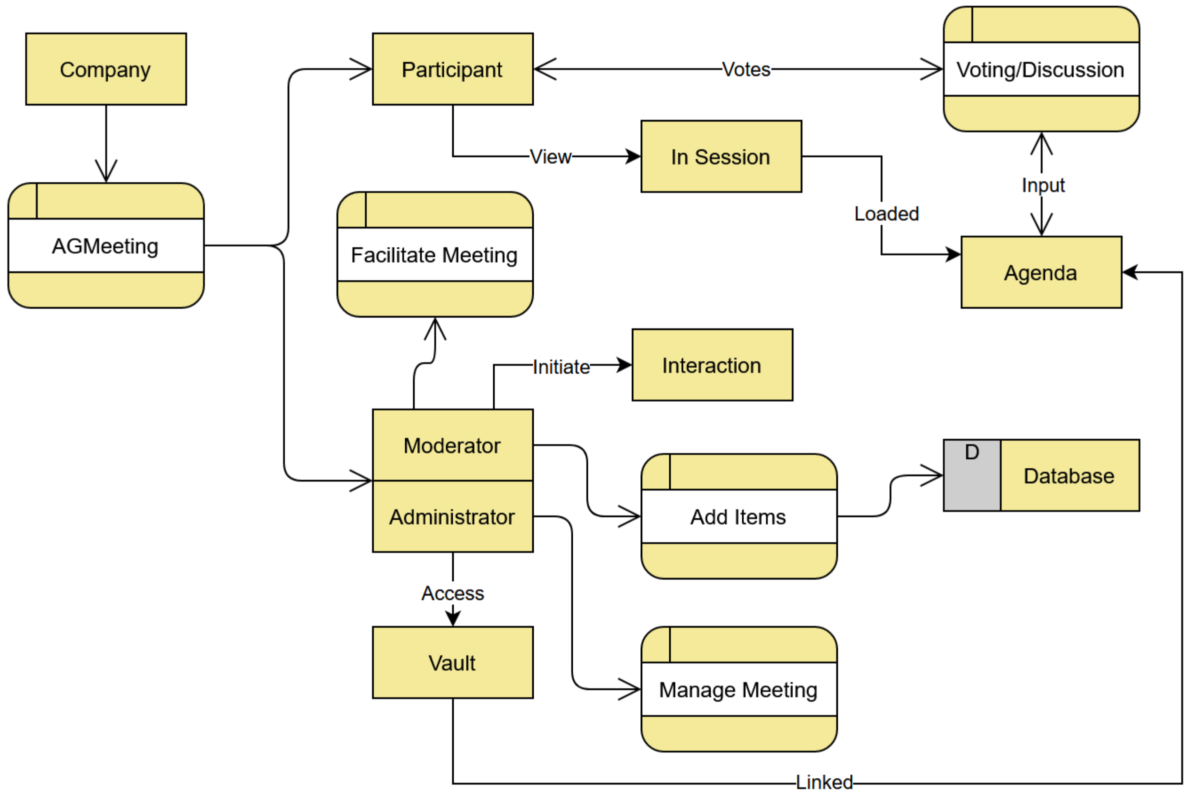
System Architecture in DFD

Level 0 Data Flow Diagram



Our level 0 data flow diagram showcases the system as a basic single process and its relation to its main external entities, which are participants, moderators, and an administrator. This is to facilitate communication during meetings.

Level 1 Data Flow Diagram



In our level 1 data flow diagram, processes within the AGMeeting system are magnified, showing how the application and its processes individually interact with external entities. A company will be interacting with the AGMeeting application, broken down into three main groups: participants, an administrator, and one or two moderators of the meeting. Participants may vote and discuss agenda items as well as view how the meeting is coming along in session. Administrators primarily manage the meeting, as they do not participate, but they as well as moderators can throw documents into a shared vault. Moderators can add agenda items, facilitate how the meeting progresses, as well as initiate interactions as part of an agenda item. In Session, Agenda, Vault, and Interaction are all pages/options to see relative information. These are all external entities where information comes and goes. Lastly, agenda items may be stored in a database. All functions an administrator has, so does a moderator.

Breakdown of the Target Components for each Milestone

This section is currently dependent on the clarification of the requirements from the client. It can be updated once the group understands what is required of the project.

Functional Requirements

<u>List of Functional Requirements</u>

1. Build AGMeeting Front-End Components & Routes

FR1: Build AGMeeting Front-End Components & Routes

<u>User Story</u> As a user, I should be able to view logs of everything that happened, access documents, have quick access to interactive events, see the schedule/catalog, be able to register as well as login.	
Requirement ID	**Requirement Definition**
FR1.0	The page should have a list view (log) of everything that has happened.
FR1.1	Document page should be present and users are able to access it.
FR1.2	Interactive buttons for quick access to functions/pages.
FR1.3	Schedule/Catalog page that is accessible.
FR1.4	Register function that allows creation of new users.
FR1.5	Login function that enables login of existing users.

**Other milestones to be determined and announced by client going forward*

Non-Functional Requirements and Environmental Constraints

1. Usability
2. Security
3. Stability

Usability

- User friendly user interface design
- Responsive and efficient application design
- Accessible via computer, tablet, or smartphone
- Should work on all internet-connected devices that have access to a browser
- Interactive buttons for mentioned functioned
- Contact functionality for issues or super administrator

Security

- User confidentiality/data integrity (especially personal information)
- Secure data submission to database

Stability

- Application reliability (especially the ability to reliably host meetings) and ironing out of bugs that may affect code safety and affect user experience

Tech Stack

- NodeJS:
    - Designed to build scalable network applications.
    - Can be used with either JavaScript or TypeScript.
    - It was designed and influenced by systems like Ruby’s “Event Machine”, and Python’s “Twisted”.
- React:
    - A JavaScript/TypeScript library that is used to build user interfaces.
    - Declarative:
        - Simple views for each state in your application, React will update and render the components as the data changes.
    - Component-Based:
        - Encapsulated components manage their own state.
        - Can keep rich data out of the DOM.
- Material UI:
    - We are using Material UI for styling the front-end of our project.
    - It is a UI library that easily interacts with React.
- PSQL:
    - A terminal-based frontend designed for PostgreSQL.
    - PostgreSQL is an open-source relational database.
    - Over 30 years of active development.
    - Has a strong reputation for being reliable, robust, and efficient.
- Redux:
    - It is a predictable state container for JavaScript apps.
    - It centralizes the application’s state.
    - Easy to test.
- Axios:
    - A promise-based HTTP client for NodeJS.
- Webpack:
    - An open-source JavaScript module bundler.
    - It takes modules with dependencies and generates static assets representing those modules.
- PassportJS:
    - An authentication middleware for NodeJS.
    - Supports SSO.
    - Supports persistent sessions.
    - Lightweight.
    - It does not mount routes in the application.

Testing

- We will implement unit testing of our code to ensure each component works as expected.
    - We will use Jest for testing out React components.
        - Jest is the most common testing framework for React projects.
        - Jest allows us to not only test the functional side of the React components but also allows us to test the usability of the website to simulate user interactions.
    - We will use Mocha for testing our NodeJS components.
        - Mocha is one of the most well know testing frameworks for NodeJS and is similar to Jest in its structure.
- We will have unit tests for each component we write, and updating of components will require these unit tests to be updated as well.
- Integration testing will be used to ensure that the React components interact with the service side of the project in order to ensure the frontend works with the backend.
- We will present the system to the client and allow them to test the entirety of the system to ensure everything works as expected.
    - It will be presented to the client at the end of the sprints the client defines to ensure that we met their expectations after completing a sprint.

Integration

- Our team will utilize Github Actions in order to ensure our code does not break when new features are merged into the main branch.
    - In Github Actions, we will implement the tests for Linux, Windows, and macOS.
- As part of the pull-request process, all tests in the project are required to pass. If a single test does not pass then merging the feature branch into main will be blocked until the failing test is investigated and fixed.

Questions and Feedback

- *What is the state of the tech stack?*

  I don’t understand this question.

- *Can you explain the relationship between your identified user groups and the various system roles identified in your DFDs?*

  Basically, companies will be using AGMeeting to host meetings, broken down into three groups, for participants, administrators, and moderators. Administrators and moderators work together to actively moderate/facilitate meetings as well as agendas, for said company.

- *What are your project milestones and what functional as well as non-functional requirements correspond to each milestone?*

  Milestones are stated under functional requirements above.

- *Are there any options for the generation and storage of user metrics being considered?*

  Not at the moment, this is not a requirement that the client has given to us.

- *What are the features that you are removing with the “Light Version” of your software?*

  The light version of the software is managed and set up by the Super Administrator, Barry, and as such we will only be working on the full version.

- *For non-functional requirements, how is your team ensuring Device Accessibility? For responsiveness, are you building in Bootstrap, or Bulma? Also, are you testing on Linux, Windows, and MacOS? We don’t see those considerations in your testing.*

  Discussed in tech stack and testing sections.

- *The unit tests are good for functional programming, but how do you plan to test the UI? Are you doing usability testing?*

  In testing section.

- There was no mention of the database being used for this project. What options have you considered for your DBMS. Also, what type of security control will the team utilize?

Currently, our only task for the project is to deliver front-end solutions. Other potential milestones have yet to be discussed.

- How will user-friendly design be ensured in your project?

To ensure a user-friendly design, we plan to incorporate design guidelines from canvas resources.

- Are there any plans for future scalability and if so how is the software going to be built to accommodate that?

Currently, there are no plans for future scalability.

- It was stated that for testing the project, you will allow the client to test the project on a regular basis; what form will this take? How can this process be conducted to exclude redundant cloning, etc.

Addressed in testing section.

- Do you have any plan to implement any tech to secure Data?

Currently, we have no plans to secure data as it is outside the scope and milestones the client has assigned us.

- The moderator and administrator roles sound very similar and the description of what they do are synonyms of each other. What can an administrator do that a moderator cannot?

The role of administrator and moderator are exactly the same as per the client. Both have access to the same level, functions and features. It is encouraged to have two people and separate the roles per their advice.