IS 6410 System Analysis & Design

Group Project

"Action Items" – Collaboration management system

Group 1

David Sarkisian

Andy Charles

Xuan Qin

Part 1: Team Organization

1.1 **Team name: Action Items**

Team members:

Team member	Skills	IT related interests
Andy Charles	Academic knowledge of Web Development, IT Security, Databases	Cloud Servers engineering
David Sarkisian	Web Development, Programming, Quality Engineering, Databases	Web Development, BI and Data Science
Xuan Qin	Academic knowledge of Networking and Servers, IT Security, and Databases	General knowledge of information systems

1.2 **Other uses for the project**

This project has not been used in any other course

Part 2: Project Selection & Requirements Analysis

2.1. Title and table of contents

Title: **Action Items**

Table of contents:

1 Team Organization

- 1.1 Team name and members
- 1.2 Other uses for the project

2 Project info & Requirements Analysis

2.1 Project title
2.2 Executive summary
2.3 Target actors
2.4 User requirements
2.5 Scope definition
2.6 Use case Diagram
2.7 Use case descriptions

3 Project Plan
3.1 Project schedules
3.1.1 Pre-implementation chart
3.1.2 Post-design chart
3.2 Effort estimates
3.2.1 Traditional estimation
3.2.2 Agile estimation

4 Analysis Documents
4.1 Logical diagrams
4.1.1 Logical ERD
4.1.2 Logical DFD
4.2 CRUD matrix
4.3 Buy vs build

5 Design
5.1 Architecture
5.2 Data storage system report
5.3 Physical models
5.4 User interfaces

6. References
7 Appendices

2.2 Executive summary

Many organizations and teams across various industries are not always efficient with the time they spend in work meetings. It's also common for teams to fail to clearly identify the purpose of the meetings and establish specific problems or goals and solutions for those as desired outcomes of the meetings. As a consequence, teams spend additional unnecessary time in meetings and sacrifice valuable time that could be spent completing actual work.

Currently there are various tools on the market designed to increase efficiency of meetings and improve the overall experience. However, most of them are missing important functionality that would allow teams to focus on specific goals and solutions for problems.

This gap can be filled by creating a platform where users are able to easily host virtual or in person meetings while focusing on clearly established agenda and specific outcomes of the meetings, additionally it would allow project leaders to set action items for each participant of the meeting to make sure all team members are involved and are contributing. Moreover, such meeting outcome data could be further integrated with project management tools and new tasks would already be assigned to appropriate team members.

Implementation of such platform could significantly reduce meeting time and increase efficiency and quality of meetings. It would allow teams more time to complete the actual work and move faster. Team members would become more involved and more likely participate in meetings and take responsibility. Finally, it would help project leaders track progress and plan better as meeting outcomes would become more predictable. A significant benefit of this platform that it could be utilize in any industry and, practically, by any team.

Great commercial profits could be generated from the platform if sold to organization using subscription model. A free version with limited features would be also offered for nonprofit organizations.

2.3 Target actors

Primary Actor 1: **Meeting owner -** this primary user of the system would be the individual who schedules and hosts the meetings. Often it will be project leader or scrum master. The meeting

owner would be also responsible for selecting the appropriate group of participants and. Establishing the agenda, and the problem or goal of the meeting. That would require the meeting owner to communicate with the rest of the team prior to scheduling the meeting. During the meeting, the meeting owner would also assign out action items to each participant.

Primary Actor 2: **Meeting participant** - this primary user is any team member who is a participant of the meeting. Such individuals are also able to assign action items to themselves or to other participants. Meeting participants are also able to leave meeting feedback.

Primary actor: **meeting spectator** – these secondary users could be participating in the meetings, but they would not be assigned and action items. They would play the role of consultants and would be able to leave meeting feedback.

Secondary actors: **non-participant team members** - these secondary users would team members who are not participating in the meeting, but action items could still be assigned to them.

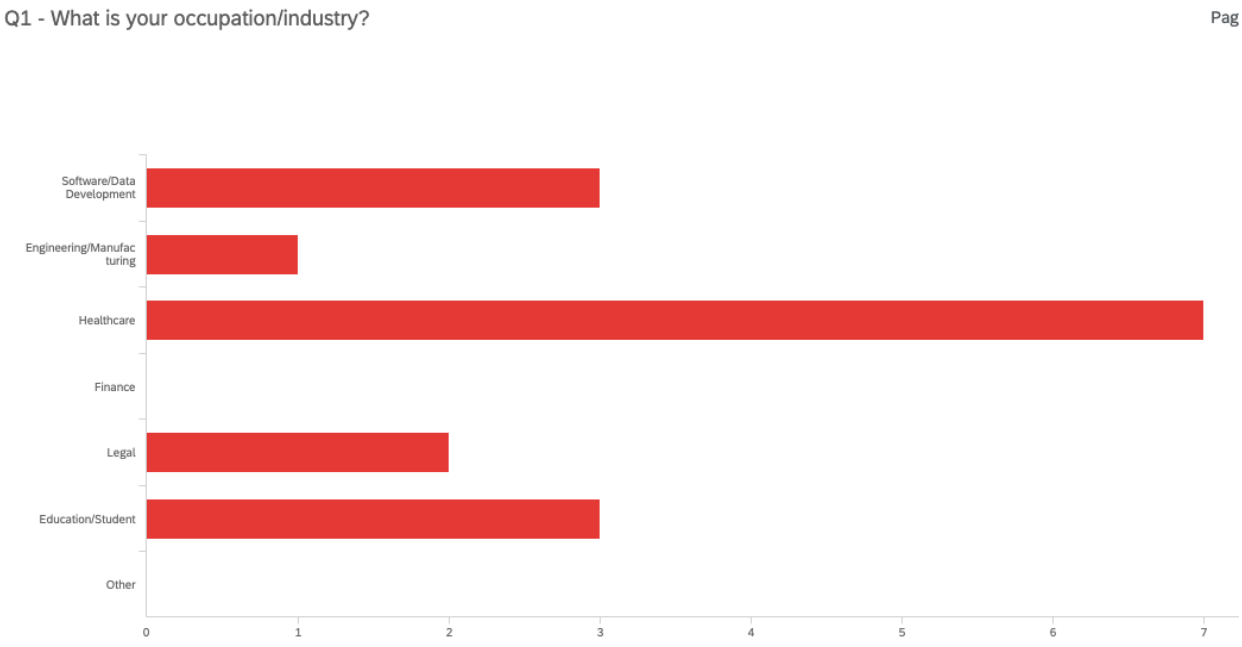
Secondary actor: **integrated information system** – these secondary users receive data created during meetings via an integration for further use or manipulation.

2.4 Requirements gathering

2.4.1 Survey

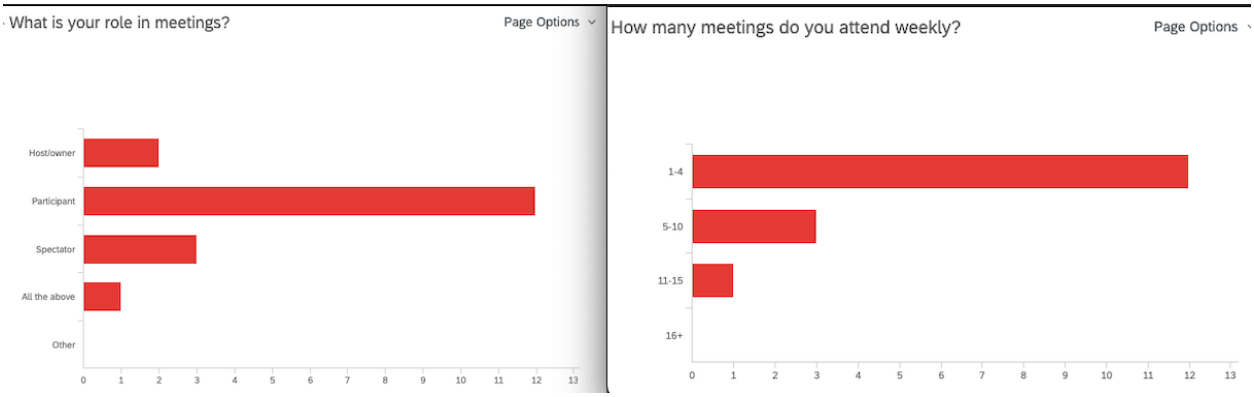
Survey was selected as the primary requirement gathering technique. Using this method allowed us to target various actor groups (with an exception of the integration information systems), and customize the questions flow with some conditional logic based on selected answers. Since the "Qualtrics" surveying software tool was used to build the survey, it was not difficult to distribute the survey. The "Distributions – Anonymous Link" feature was used to complete survey distribution. The link was sent to individuals with various occupation industry and team role to gather data from a diverse group of potential actors. Qualtrics also allowed us to easily analyze survey results using features under the "Reports" tab.

The survey includes the total of 20 questions, but depending on the answers, some users will not see all the questions as there is conditional logic implemented.
The survey does not include any personal information questions, such as names, places of work or age, as those questions are irrelevant for our project. It also allows users to remain anonymous and make the surveying experience seamless.
The first multiple-choice question “What is your occupation/industry?” collects general information about the industries in which most of our primary and secondary actors work. This allowed us to identify the potential market opportunities for our product.



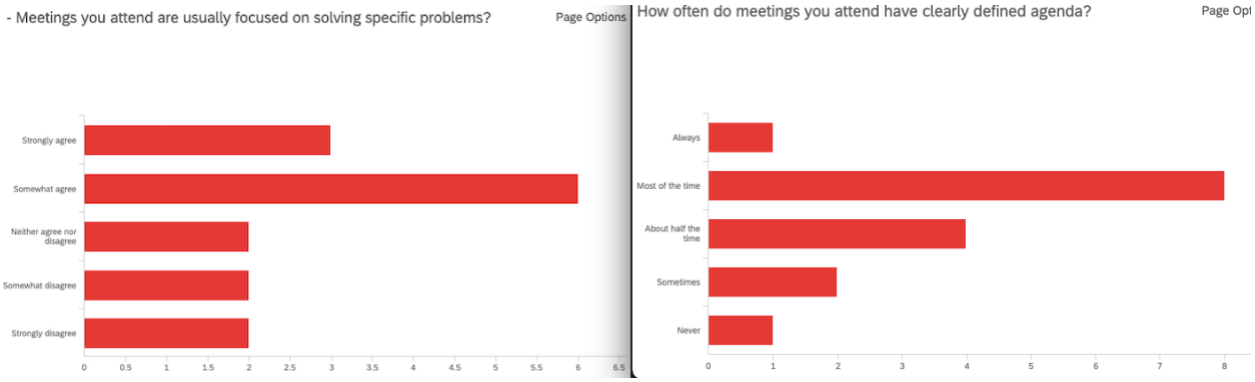
The diagram above shows that most users work in healthcare industry, but there are also plenty of answers falling into other categories such as software development, engineering, legal and education. This indicates that our product could have potential market opportunities across multiple industries.

The initial question is followed by questions collecting information about frequency of meetings and user’s role in meetings. The vast majority of users attend 1-4 meetings per week, while most of the users are meeting participants a few hosts and spectators.



The question “Do you perform leadership duties within your team?” is used to implement conditional logic and display certain questions for team leaders. The results show that about 56% of participants are leaders.

Survey presents a few important questions collecting data about meeting experience for the users. Despite the fact that the results show that most users agree that meetings usually have agendas clearly defined as well as they focus on solving specific problems, we still believe that the most teams face meeting productivity issues. The results may have been slightly inaccurate because of how they have been structured. Similar results are seen for “If you are participating in a planning meeting, how often is it clear what work has been assigned to you?” question.



Following several questions were added to the survey to collect data that would help decide if specific features need to be offered withing the product. These features are:

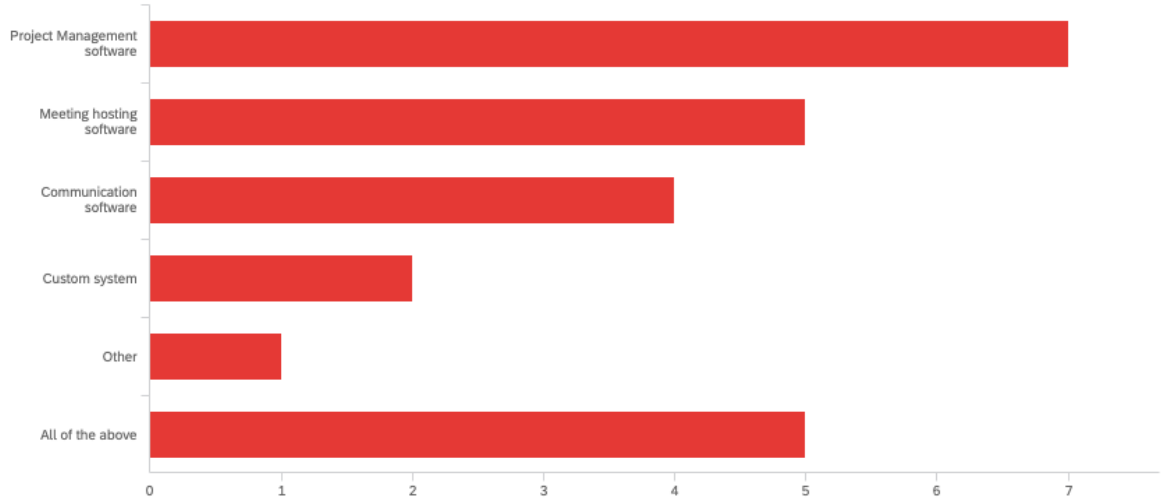
- Display meeting types: 87% users answered ‘Yes’ or ‘Maybe’ – feature included. Additionally, a question inquiring about what meeting types users participate in was added with a variety of options and evenly distributed results validated our decision to include this feature in the product.
- Ability to assign specific tasks to team members for team leaders: 90% users answered ‘Yes’ – feature included
- Ability to receive meeting feedback from meeting participants: 69% users answered ‘Yes’ – feature included
- Display remaining meeting time: 94% users answered ‘Yes’ – feature included. (Additionally, about 88% users state that meetings tend to go overtime.)
- Host polls in meetings: 56% users answered ‘Yes’, 25% users answered ‘Maybe’ – feature included.
- Support spectator user type: 75% users answered ‘Yes’ – feature included

The survey also contains questions targeted to identify the extent to which our product should be integrated with other systems. About half the users answered that their team uses additional tools to document meeting results. Most of those users then think that such tools are effective. However, about 93% users answered ‘Yes’ to the “Would you want to be able to import your documented meeting results into other platforms?” question, so we decided to include this type of integration.

Additional integration feedback was gathered from the answers to “Which platforms would you want your meeting software to integrate with?” question. We decided to provide a universal option to export meeting data from our tool in various formats and the ability send such data via the API.

Q17 - Which platforms would you want your meeting software to integrate with?

Page Options



#	Field	Choice Count	
1	Project Management software	29.17%	7
2	Meeting hosting software	20.83%	5
3	Communication software	16.67%	4
4	Custom system	8.33%	2
5	Other	4.17%	1
6	All of the above	20.83%	5

Other

MS Word

Each user is asked to provide any additional feedback at the end the survey. The users did not leave any useful feedback in this section.

2.4.2 Research

Research was also selected as a requirement gathering technique in addition to Surveying. This type of requirements gathering covers all of the target actors listed in section 2.3. It also appeared to be the best fitting requirement gathering method to use for integrated information systems actors as such actors cannot be interviewed or surveyed.

After searching the internet, following facts finding tasks were set:

- Identify current competing products on the market with shared target customer base
- Identify features offered by competitors that bring additional value to the product
- Identify current trends/news among largest players on the market.

Gathering such facts and information helped us identify areas of improvement, consider adding more features to the system.

It is important to note that there are not that many products on the market that offer the all-in-one software tool for hosting meetings with defined agenda and documenting task assignments associated with the agenda. We were able to, however, identify some gaps in the product that could be filled by developing additional features. One of such features is integrating “Action Items” into Jira prior to meeting. Integration interface could be added directly within a Jira “Epic” so that teams could easily complete planning meetings and then import action items back to Jira as tickets.

After finding multiple competing products, we selected two most powerful ones: Hive and Flock. A great advantage of Hive is that it’s a standalone project management platform, but also offer a wide variety of integration with tools such as Slack, Zoom, Jira, MS Teams and more. The product offers various collaboration features, some of which our product lacks. In order to keep our system competitive, we decided to adopt some of them: meeting templates and notes https://hive.com/product/. Templates allow to preload meetings agendas and default action items leaving little input for the user to add. Notes are additional input fields for action items that would provide more detail for the task.

Flock is another competing product that offers a very similar set of features. Although, there is one that could put them ahead of the "Action Items", and that feature is Reminders. It allows users to choose deadlines for tasks, and remind user about them by the bot https://www.flock.com/features/productivity/todos/.

2.5 High-level scope definition

2.5.1 Functional User stories:

1) As a **meeting owner**, I want to **create a meeting and set meeting type** so that **all meeting participants are able to view it**.

Acceptance Criteria:

- Meeting owner is able to create meeting and set meeting type
- Meeting participants and meeting spectators are able to join the meeting and view the meeting type
- Meeting can be created from a template

2) As a **meeting owner**, I want to **set meeting agenda** so that **all meeting participants are able to view it**.

Acceptance Criteria:

- Meeting owner is able to set meeting agenda
- Meeting participants and meeting spectators are able to view the meeting agenda

3) As a **meeting owner** or **a meeting participant**, I want to **set action items to meeting participants and team members** so that **they understand the work that they need to complete in the future**.

Acceptance Criteria:

- Team members names are listed on the screen in the "action items" view
- Users are able to input, assign, and view action items

4) As a **meeting spectator**, I want to **leave feedback for the team after the meeting**, so that **the quality of work can be improved**.

Acceptance Criteria:

- Meeting spectators are able to leave feedback notes for the team

5) As a **user**, I want **to be able to be able to vote in a poll** so that **my input can be considered**.

Acceptance Criteria:

- Meeting owner can create multiple polls in a single meeting
- Meeting participants can vote in polls

6) As a **meeting owner or participant**, I want **to leave additional action items notes** so that the team may review them and get additional details for discussed topics.

Acceptance Criteria:

- Users are able to leave meeting notes for action items

7) As a **non-participant team member,** I want to **receive reminders** so that **I complete the assigned task on time due.**

Acceptance Criteria:

- Team members receive action items reminders

8) As an **integrated information system**, I want **to receive meeting data from "Action items" via an API** so that **the data can be easily accessed in the system.**

Acceptance Criteria:

- Meeting result data can be exported from the system
- Meeting result data can be imported into a different system

2.5.2 Non-functional user stories

9) As a **user**, I want **the system to be able to scale quickly so that up to 250 team members can input data simultaneously without system performance decreasing.**

Acceptance Criteria:

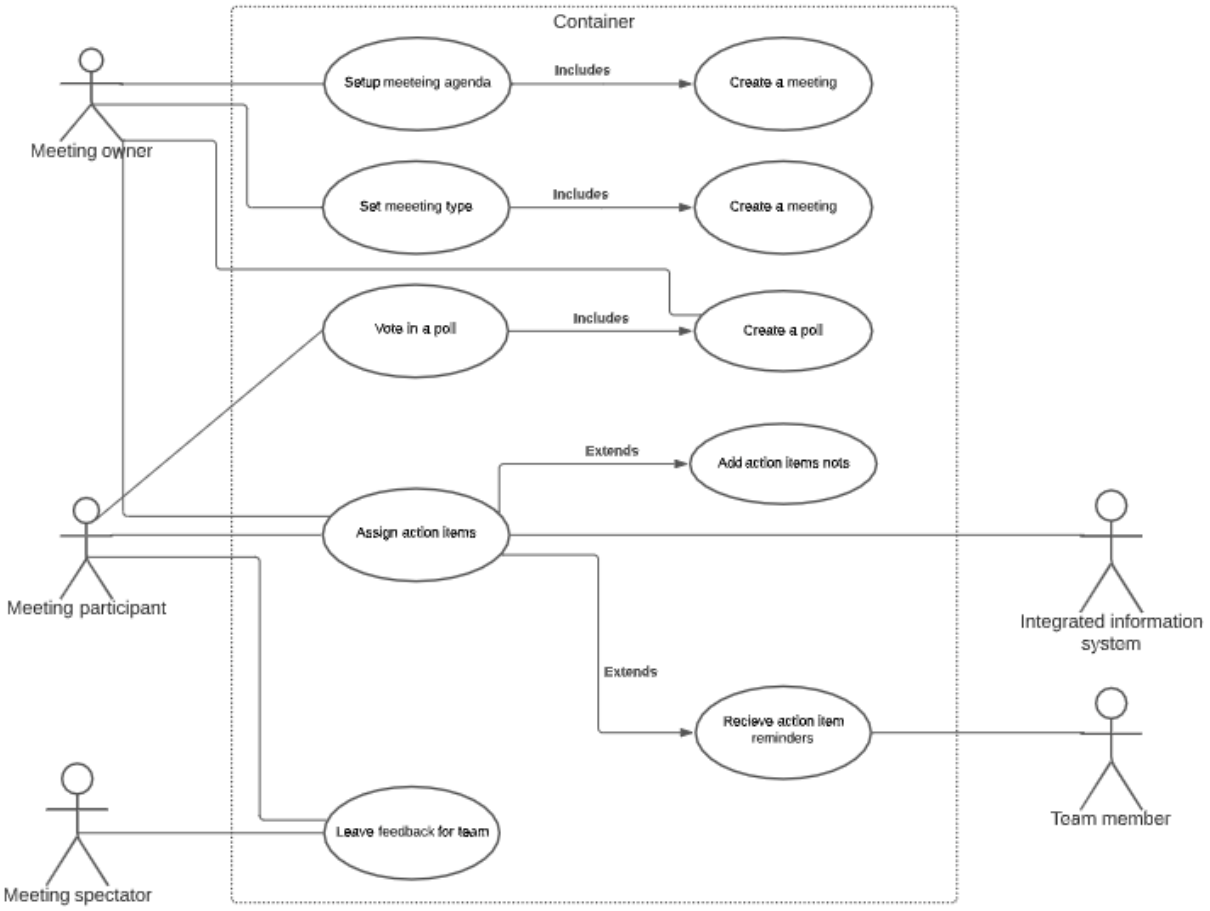
- System performance is not decreasing with more increasing user input

10) As a **user**, I want **the action items data to be only available to users with granted access**, so that **unauthorized users are not able to access it.**

Acceptance Criteria:

- Authorized users are able to access action items data
- Unauthorized users are not able to access action items data

2.6 Use Case Diagram



2.7 Use case narratives

Use Case 1	
Description	Meeting owner can click to add meeting agenda
Actors	Meeting owner
Pre-Condition	Meeting owner wants to set meeting agenda
Post-Condition	Meeting agenda is set and display at the top of the action block
Basic Success Flow	
1. Meeting owner clicks on the add meeting agenda 2. A successful message shows that the meeting agenda is displayed at the top of the action block	
Alternative Path	
A1.	

Use Case 2	
Description	Meeting owner can click to add action item
Actors	Meeting Owner
Pre-Condition	Meeting owner wants to add action item
Post-Condition	Action item is added next to each participant
Basic Success Flow	
1. Meeting owner clicks on the add action item button 2. Green check mark shows that the action item has been added to each participant	
Alternative Path	

Use Case 3	
Description	Meeting spectator can click to add feedback
Actors	Meeting spectator
Pre-Condition	Meeting spectator wants to leave feedback
Post-Condition	Feedback/action item is added next to each participant
Basic Success Flow	
1. Meeting spectator clicks on the add feedback button 2. A green check mark shows that the feedback is successfully added to each participant	
Alternative Path	
A1: Noting is written in the feedback box 1. An error pops up saying that the fled is empty	

Use Case 4	
Description	Meeting participant can click to display the poll question to vote
Actors	Meeting participant
Pre-Condition	Meeting participant wants to vote
Post-Condition	The poll questions are selected and ready to submit
Basic Success Flow	
1. Meeting participant clicks on the poll questions 2. Successful message is displayed once an answer is selected and submitted	
Alternative Path	
A1. Meeting participant doesn't select an answer 1. Error message pops up saying that an answer must be selected	

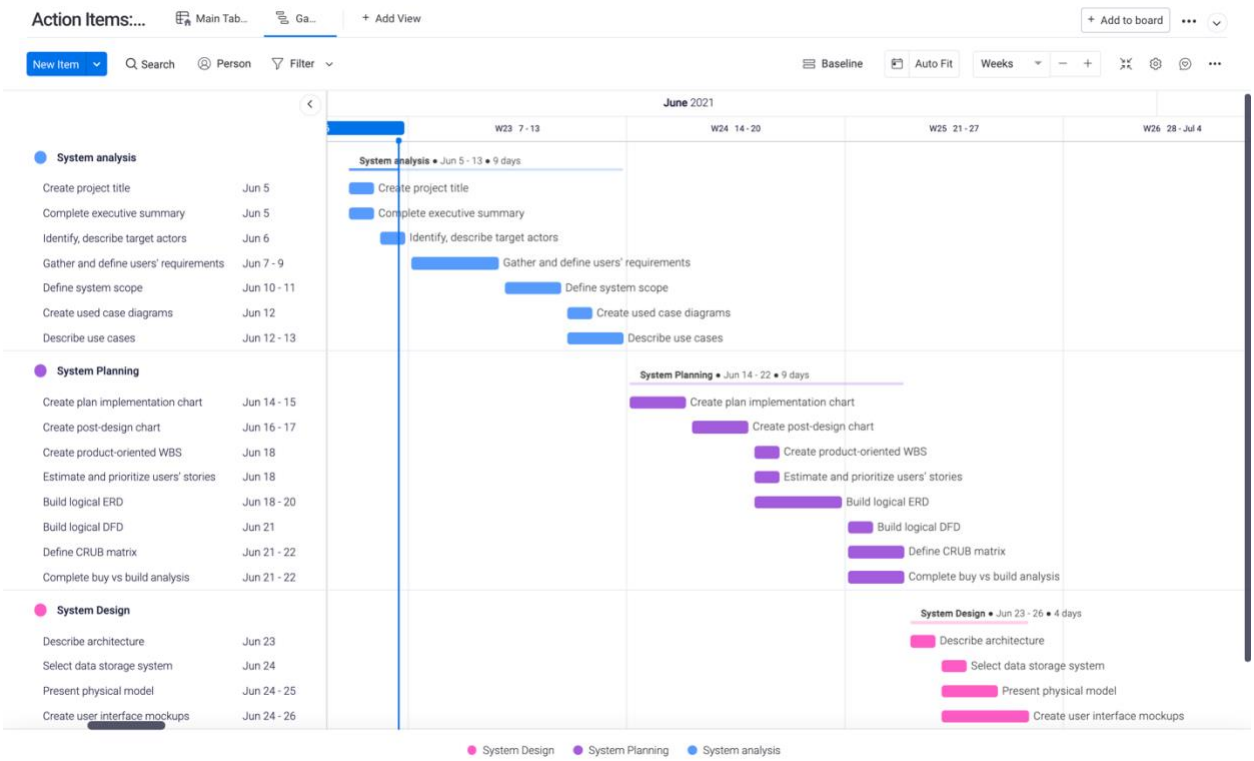
Use Case 5	
Description	Meeting participant can click on display button to display poll question on the screen
Actors	Meeting participant
Pre-Condition	Meeting participant wants to create suggestions
Post-Condition	Suggestions are created and added to the voting result
Basic Success Flow	
1. Meeting participant clicks on the poll suggestion box to add suggestion 2. Successful message is displayed once a suggestion is made and submitted	
Alternative Path	

Use Case 6	
Description	Non-participant team member can click on the reminder button to set reminders
Actors	Non-participant team member
Pre-Condition	Non-participant team member wants to receive action item reminders
Post-Condition	Reminders are set for all action item
Basic Success Flow	
1. Meeting participant and meeting owner click on the reminder button to set reminders	
Alternative Path	
A1: Reminder is not set 1. An error message pops up and saying that a reminder needs to be set before assigning action item	

Use Case 7	
Description	When action item is added, data are transferred to the integrated information system
Actors	Integrated information system
Pre-Condition	Integrated information system wants to receive data from action items
Post-Condition	
Basic Success Flow	
1. Users can successfully access the data	
Alternative Path	

Part 3: Project Plan

3.1.1 Pre-implementation GANTT chart:



3.1.2 Post-design GANTT chart:

Task Mode	Steps	PERT Estimate	High cost %	Low Cost %	Additional Cost	Total Cost Estimate	Cost	Duration	Start	Finish	Pre
1	Implement User Interface	35.7	%75	$25	$	$4,480.00	$4,560.00	6 days	Tue 7/13/21	Tue 7/20/21	
2	Create Meeting Database	20	%85	%15		$3,360.00	$3,360.00	3 days	Wed 7/21/21	Fri 7/23/21	1
3	Perform User Testing	24	%15	%85	$500.00	$4,480.00	$4,320.00	6 days	Mon 7/26/21	Mon 8/2/21	2
4	Create Wireframe	15	%10	%90		$2,240.00	$2,240.00	2 days	Tue 8/3/21	Wed 8/4/21	3
5	Total	94.7 Hours 20 days				$14,980.00	$14,560.00	0 days	Wed 7/28/21	Wed 7/28/21	

3.2 Software development estimate

Steps	Worst Case Hours	Best Case Hours	Likely Case Hours	PERT Estimate
Implement User Interface	40	30	36	35.7
Create Meeting Database	25	16	20	20
Perform User Testing	30	24	23	24
Create Wireframe	18	13	15	15

Formular: Worst case + Best case + 4x Likely case/6

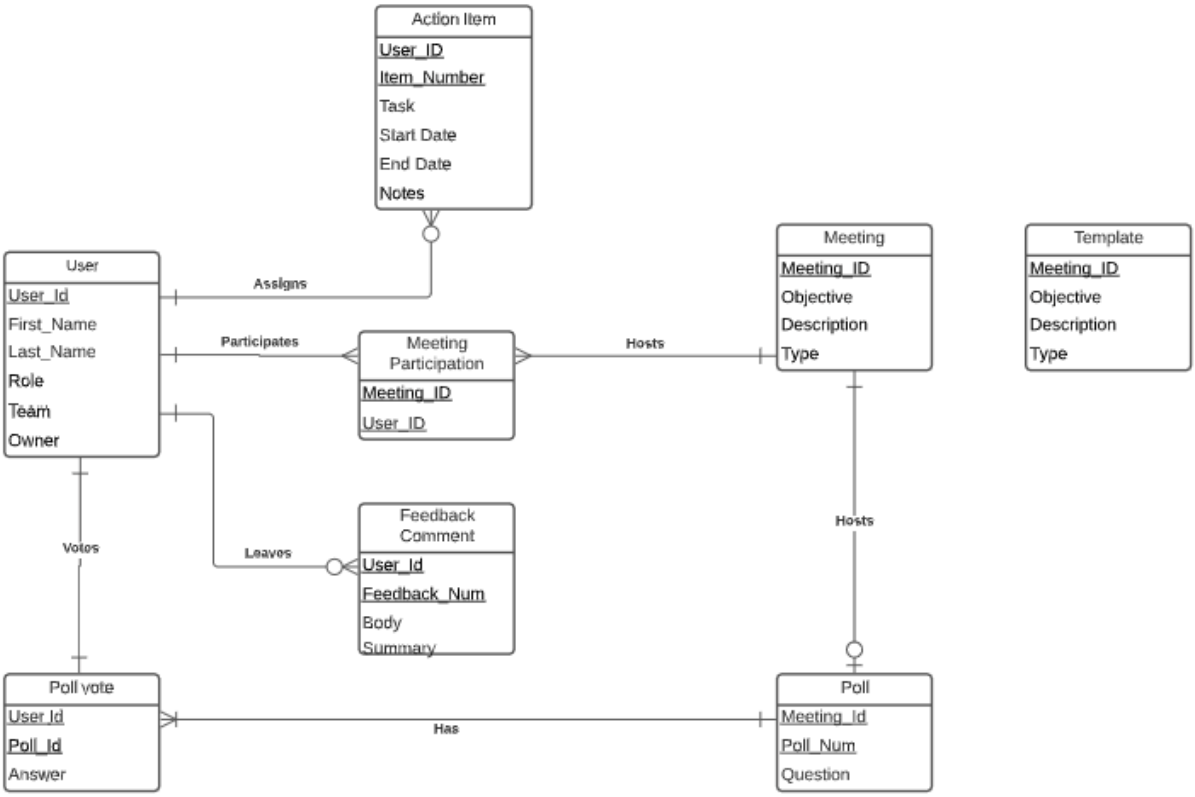
We estimated the high cost at $50 per hour and the low cost at $45 per hour with a total cost of 14,980 dollars over 20 days and 94.7 hours for the implementation of this project.

Steps	PERT Estimate	High cost %	Low Cost %	Additional Cost	Total Cost Estimate
Implement User Interface	35.7	%75	$25	$	$4,480.00
Create Meeting Database	20	%85	%15		$3,360.00
Perform User Testing	24	%15	%85	$500.00	$4,480.00
Create Wireframe	15	%10	%90		$2,240.00
Total	94.7 Hours 20 days				$14,980.00

Part 4: Analysis Documents

4.1.1 Logical ERD

4.1.1.1 Diagram



4.1.1.2 Metadata

Entities

User – this entity represents the user interacting with the system. It holds attributes identifying and describing the user. The attributes include user's id, first name, last name, role, team and if the user is owner or not. The user id is a numeric value that must be unique.

Meeting – this entity represents the meeting itself. Objective attribute represents the summary of the problem or agenda needed solved as the outcome of a meeting. It also includes type and description attributes used to provide detailed information about the meeting purpose. Meeting id must be unique.

Meeting participation – this entity reflects a log of a user participating in a meeting. It includes meeting id and user id.

Action Item – this entity represents a task assigned to a user during a meeting. The user id and item number are used to uniquely identify the action item. Task attribute hold the value of the item's summary. Start date, end date, and notes are attributes used to describe the action item and provide more detailed information about it.

Feedback Comment – this entity represents a feedback comment that user may leave for the team. It references user id as the comment is tied to a user. Feedback number attribute is used to help identify a specific comment. Summary and body are the attributes used to hold the content of the feedback comment.

Poll – represents an optional poll that can be help within a meeting. It references meeting via meeting id, adds a poll number for identification and holds the question value as question attribute.

Poll Vote – represents user's vote in a poll. References the user and poll through their respective ids and stores the answer.

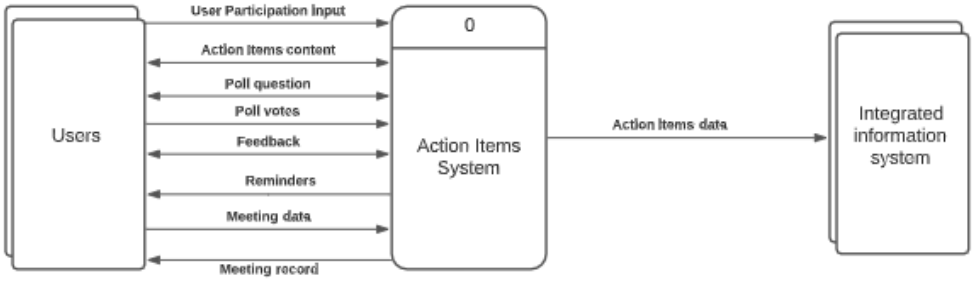
Template – this entity holds hard coded meeting data. It can be used to quickly prefill meeting data.

Relationships and cardinalities

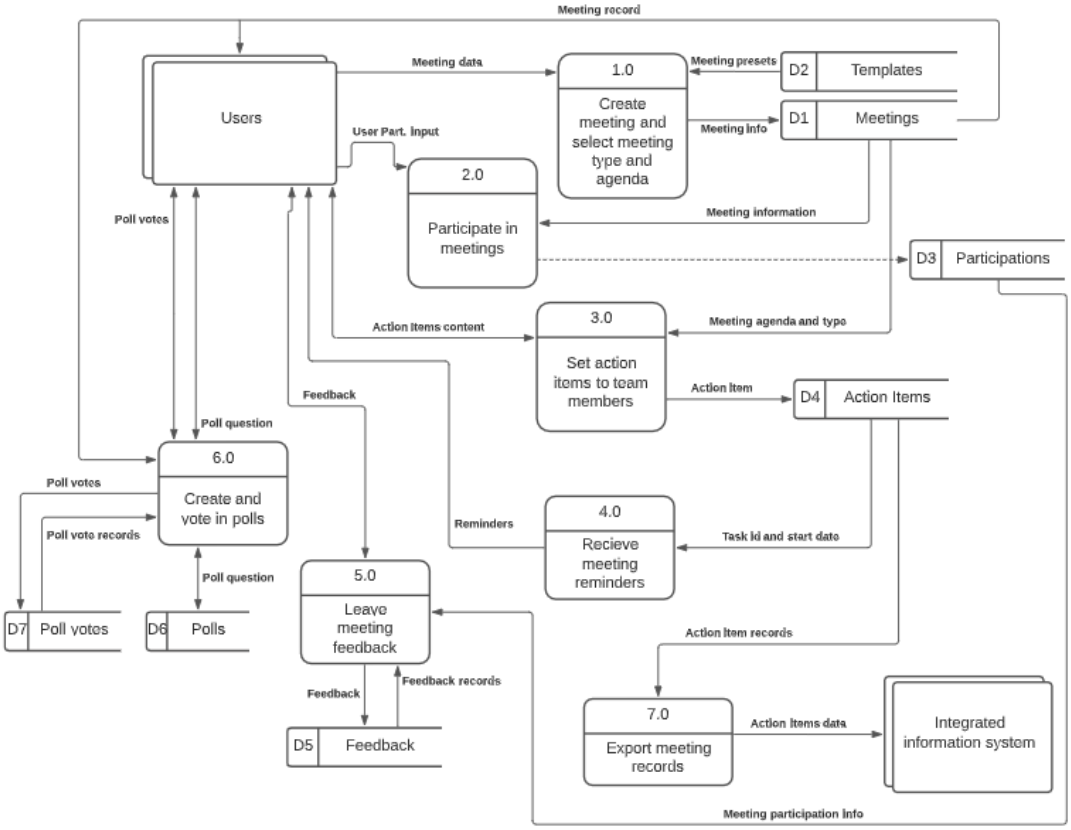
- A user may assign none or many action items
- A user participates in one or many meetings
- A meeting hosts one or many participations
- A user may leave many comments
- A user may vote in a poll
- A meeting may host a poll

4.1.2 Logical DFD

4.1.2.1 Context Level



4.1.2.2 Level 0



4.2 CRUD matrix

ID	Process	Meetings	Meeting participations	Templates	Action items	Feedback	Polls	Poll votes
1.0	Create meeting and select meeting type and agenda	C		R				
2.0	Participate in meetings	R	C					
3.0	Set action items to team members	R			CRUD			
4.0	Recieve meeting reminders				R			
5.0	Leave meeting feedback		R			CRU		
6.0	Create and vote in polls						CR	CR
7.0	Export meeting records				R			

3.3 Buy vs build analysis

We need to have a database as part of our product to manage the information of meeting owner(s), meeting participants, non-participant team members and meeting spectators. We also need to manage the information of meeting progress, milestones with timestamps, and send out notifications when there is an action item due soon.

We will consider buying a premade database service and integrating with our app or building our own in-house data storage and management system. We decided to compare building vs Hive/Flock as those are the software tools currently on the market that offer the all-in-one tools.

Building in House	
Pros	**Cons**
We can create a **fully customized** data storage solution. Can create a database that is fully integrated with our app from the ground up. We can update and change our data storage based on the specific requirements of our app and users.	The security features are not going to be as robust in an in-house solution. We will have to constantly **maintain** and update our integrations which risks mistakes which can cause problems, and the users' information could be jeopardized.
We can build up our employee's **institutional knowledge** and be able to troubleshoot any problems with people that know the software best.	The **Costs** are much more variable as we don't know the exact time or money it will take to build a storage system from scratch, and usually it's more costly than buying.

Buying off the shelf	
Pros	**Cons**
Have **Costs** that are set and consistent for the full length of our usage. The templates and notes of Hive are free to use and can be integrated with Slack, Zoom, Jira, MS Teams and more.	Hive or Flock is not designed well for the large amounts of data we are collecting, especially if we start collecting unstructured data. We could be more **efficient** with a customized in-house build.
We can **focus** on our profit generating activities without spending time building a data storage and management solution that isn't worth it. More time can be spent working on our actual features, allowing the preloading of action items.	Data will be stored on external software and we will have to **trust** that component's security protocols without being able to test it.

Part 5: Design Documents

5.1 Architecture overview

5.1.1 Supported platforms

"Action items" will be used by collectives in professional or academic settings to accomplish maximum efficiency in meetings and collaboration sessions. Also, the tool's user interface requires a large amount of screen space to provide best user experience. Considering these factors and the estimated cost to build mobile and desktop versions of the application, it was decided that it is best not to support "Action Items" on mobile and desktop platforms and offer the product exclusively on web browser platforms.

5.1.2 Architecture objectives

In order to successfully deploy and maintain "Action Items" in production environment and fulfill functional and non-functional requirements of the system, we identified following objectives for our architecture design:

- Provide experience with easily identifiable data entities for users to input and read;
- Perform application logic, such as mapping action items to respective users;
- Store data entities for short- and long-term accessibility;

- Obtain secure, scalable, and reliable hosting for the front-end web application, backend service and the database;
- Obtain and reliable hosting for the source code repository and integrate with the site hosting platform for continuous deployment and testing;
- Optimize costs of the hosting systems;
- Export and transfer data entities to other systems;

5.1.3 Architecture design

After analyzing functional and non-functional requirements and the architecture objectives of the system, we decided to use object-oriented approach with MVC framework. Using the OOP paradigm allows for easier development while storing, transferring, and presenting data entities in a more intuitive manner. The presentation logic will be handled using a popular JavaSctipt framework, such as ReactJS. The application logic will be handled using a programming language supported within MVC framework, such as Ruby and it's MVC framework – Rails. The data access technology will be discussed in item 5.2. All different components will communicate with each other vie REST API.

5.1.4 Hosting platforms

We selected a combination of existing popular tools used for hosting web applications. One of the AWS products we decided to utilize is AWS Amplify. This product appeared to be a great fit for our objectives as it provides strong security and availability for hosted applications, allows to scale on demand, allows to host both front and back-end services written in various languages, easy to integrate with other AWS products, source code repository hosting platforms, and other external platforms. We will be hosting our front-end application and back-end service on AWS Amplify.

We selected GitLab as our source code repository management system. It comes from a reliable provider, offers strong security, availability, and user experience. It also allows to implement continuous integration and run unit and integration tests per commit. AWS Amplify and GitLab can be easily integrated and new features can be automatically deployed to production environment.

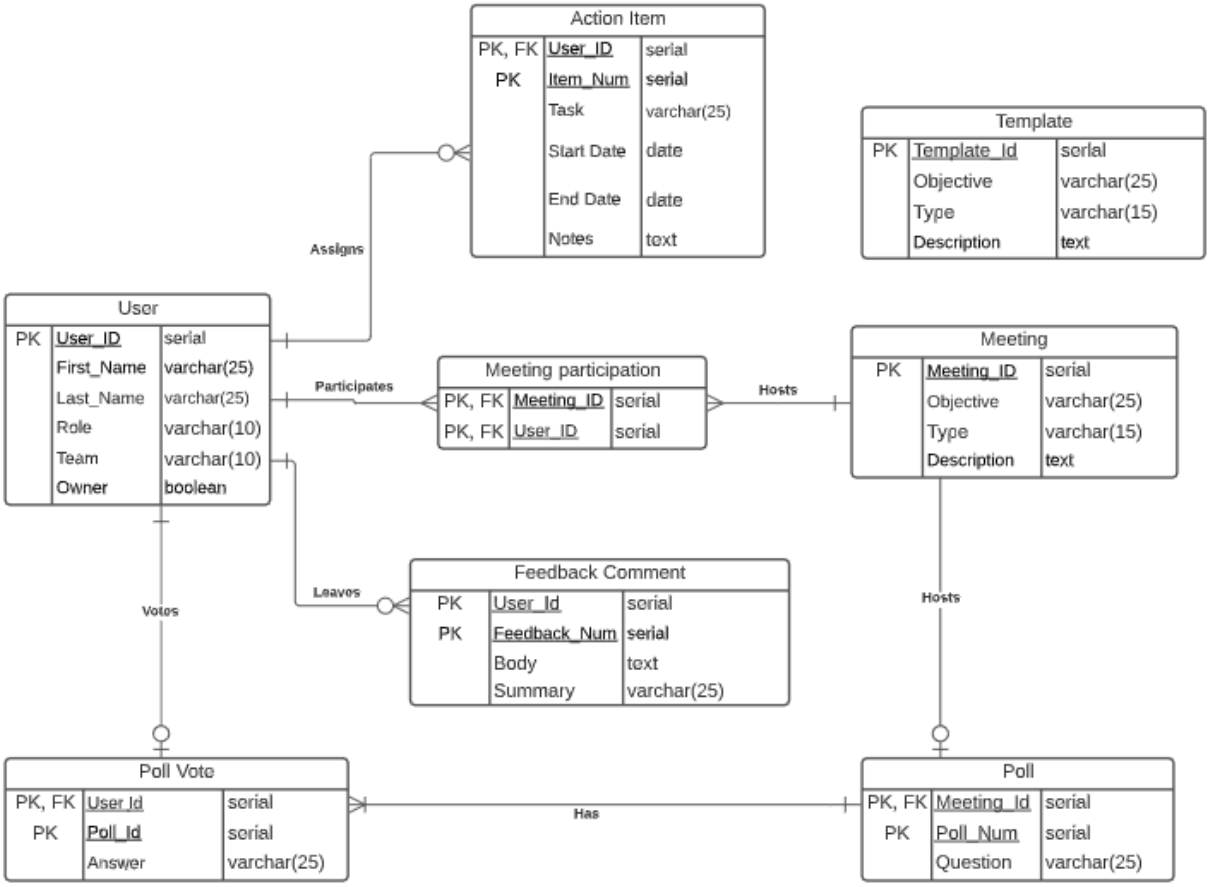
Reminders will be sent via email using AWS SES (Simple Email Service Classic)

5.2 Data-storage

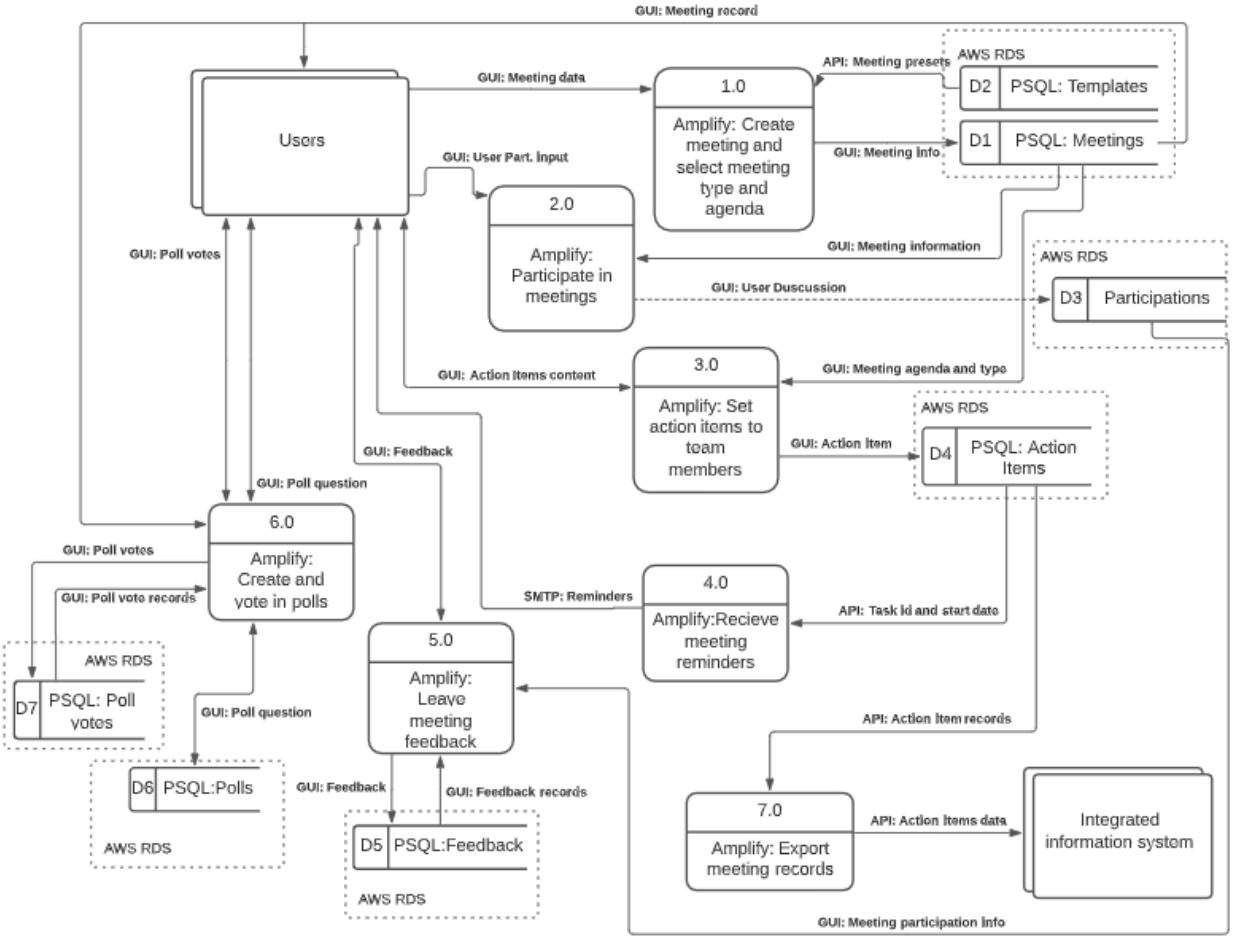
To store our data entities, we decided to use a relational database. There are various popular tools that support such databases and it plays well with the object-oriented design of the application. PostgreSQL is a great database system as it is open source, and is known for its reliability and security. The database will be hosted using AWS RDS as it is a great tool to setup, operate and scale a relational database in cloud-environment and can be integrated with Amplify.

5.3 Physical models

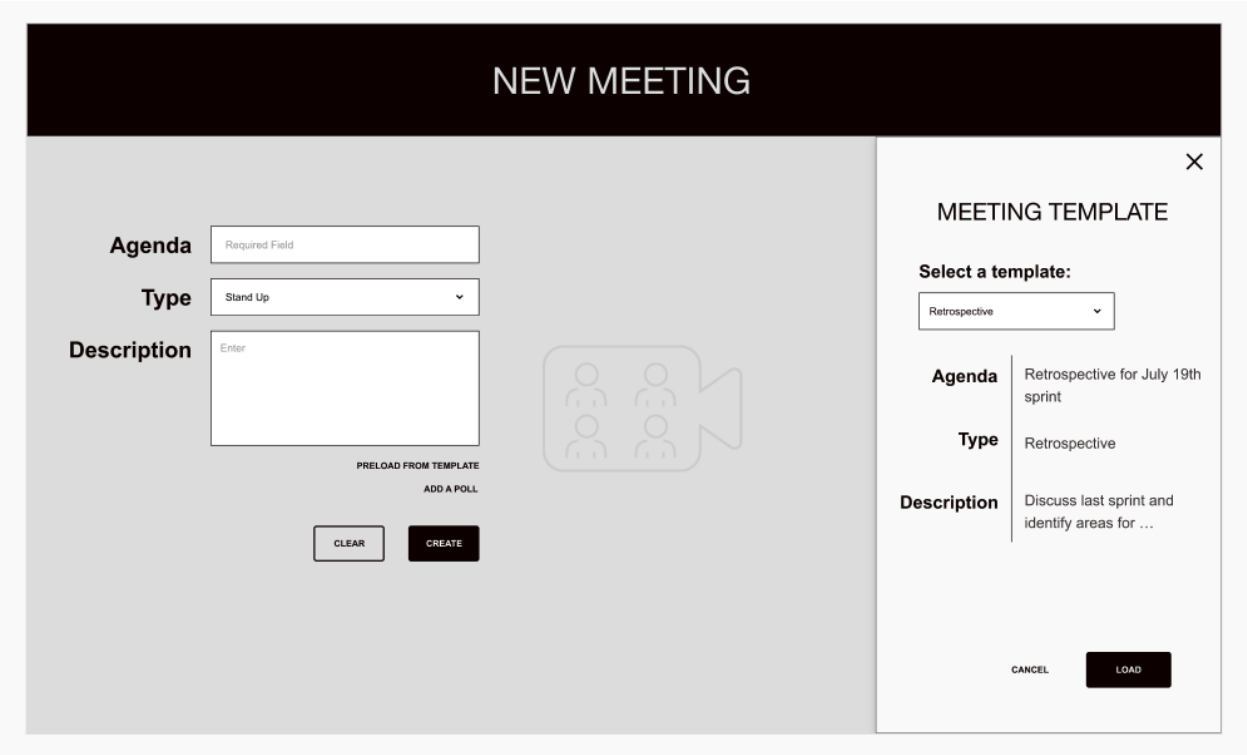
5.3.1 Physical ERD

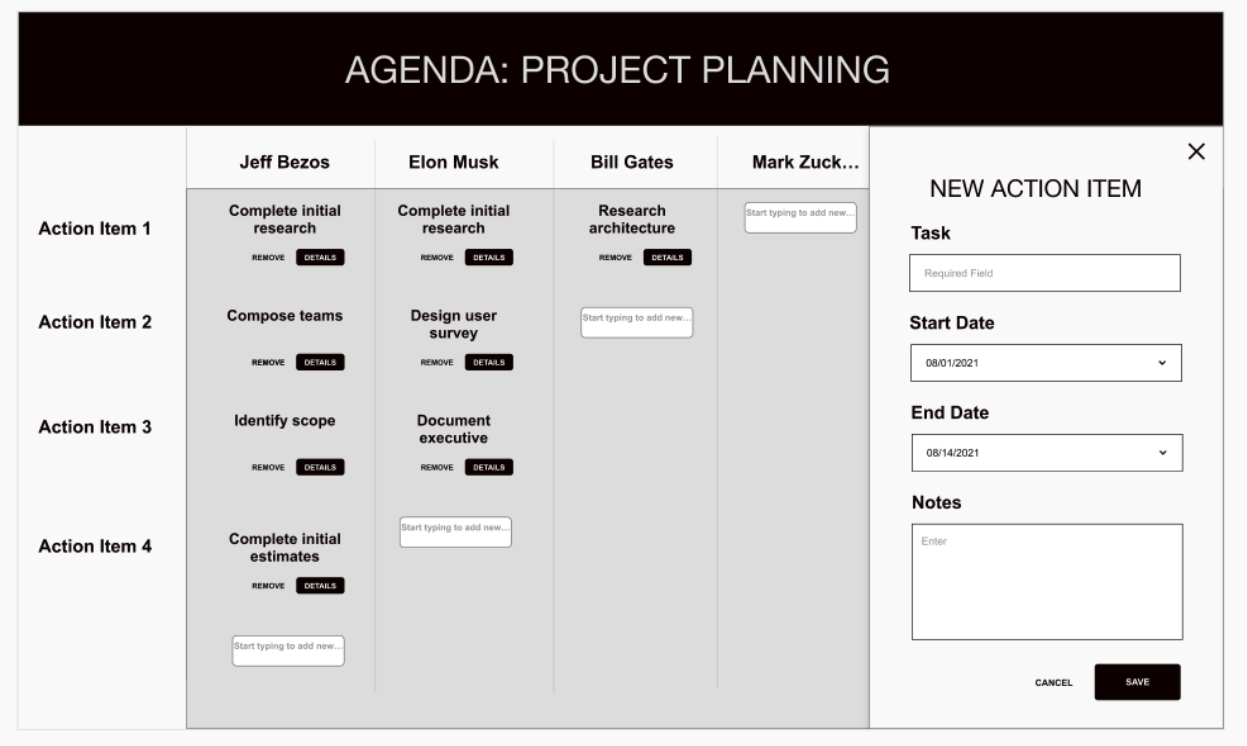


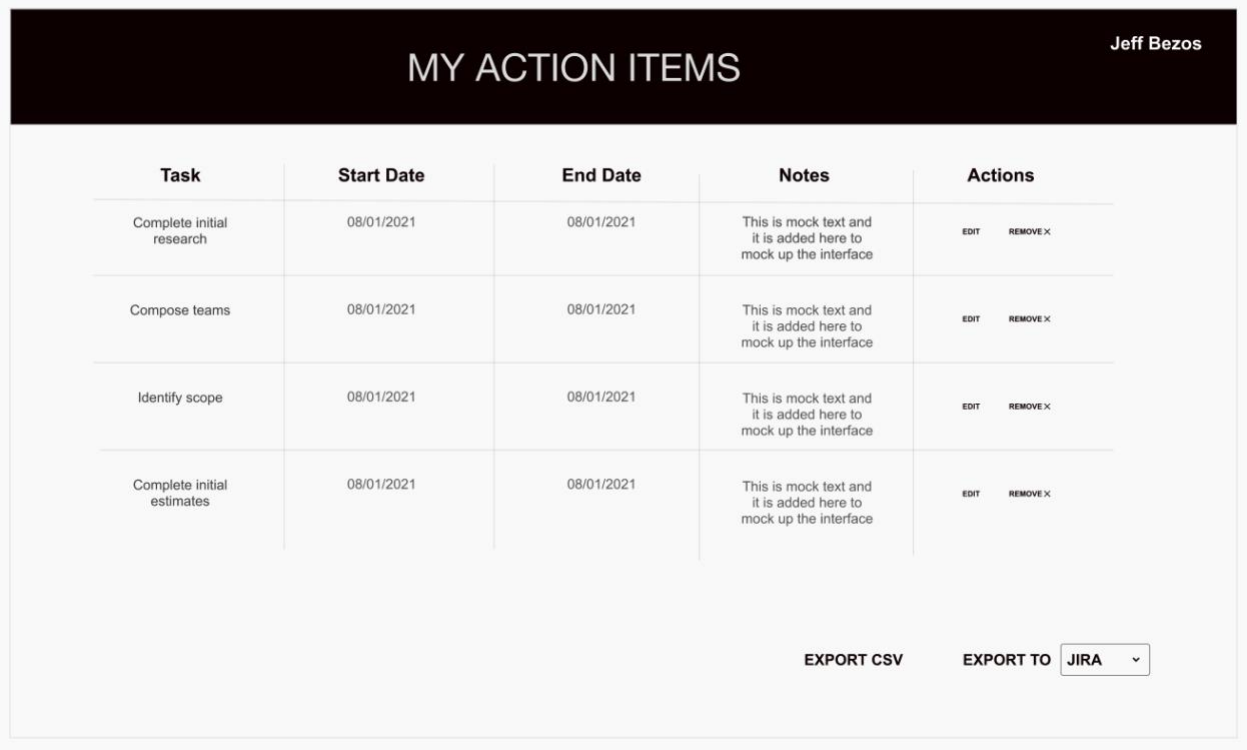
5.3.2 Physical DFD



5.4 Mock interfaces







Part 6: References

6.1 References:

PostgreSQL information: https://www.postgresql.org/

AWS Amplify information: https://aws.amazon.com/amplify/faqs/

AWS RDS information: https://aws.amazon.com/rds/faqs/

GitLab information: https://about.gitlab.com/

6.2 Software used

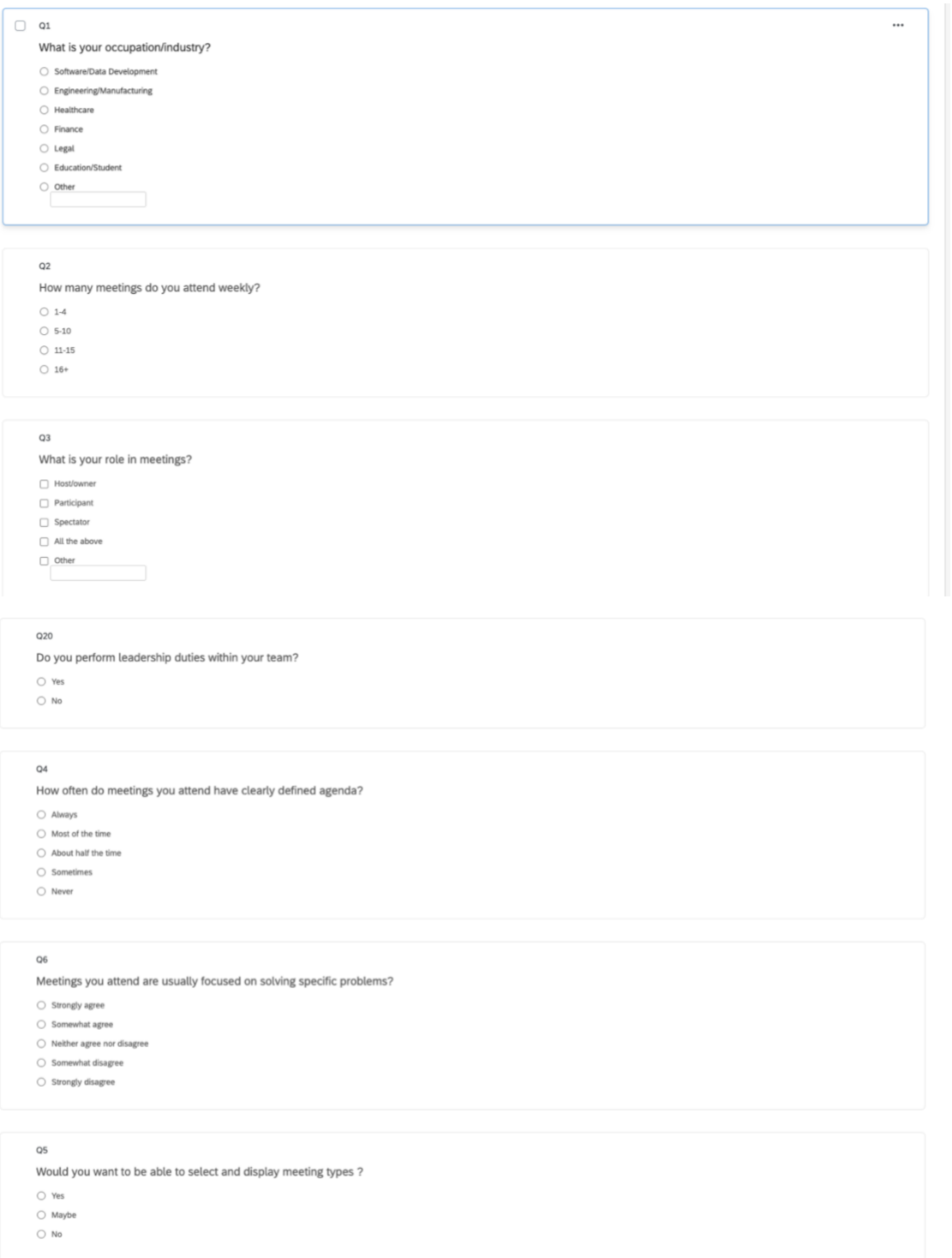
Survey – Qualtrics

Diagramming – Lucidchart

Mockups – Adobe XD

Part 7: Appendices

2.4.1 – Full Survey



Q1

What is your occupation/industry?

- Software/Data Development
- Engineering/Manufacturing
- Healthcare
- Finance
- Legal
- Education/Student
- Other

Q2

How many meetings do you attend weekly?

- 1-4
- 5-10
- 11-15
- 16+

Q3

What is your role in meetings?

- Host/owner
- Participant
- Spectator
- All the above
- Other

Q20

Do you perform leadership duties within your team?

- Yes
- No

Q4

How often do meetings you attend have clearly defined agenda?

- Always
- Most of the time
- About half the time
- Sometimes
- Never

Q6

Meetings you attend are usually focused on solving specific problems?

- Strongly agree
- Somewhat agree
- Neither agree nor disagree
- Somewhat disagree
- Strongly disagree

Q5

Would you want to be able to select and display meeting types ?

- Yes
- Maybe
- No

Q7

What types of meetings do you usually host/participate in?

- [ ] Stand Up / Team update
- [ ] Planning
- [ ] Retrospective
- [ ] Collaboration
- [ ] One on One
- [ ] Informational / Presentation
- [ ] Other
- [ ] All of the above

Q8

Display this question

If What is your role in meetings? Host/owner Is Selected

Or Do you perform leadership duties within your team? Yes Is Selected

As a team leader or meeting host, would you want to be able to assign specific tasks to the team members/participants?

- ○ Yes
- ○ No

Q9

Display this question

If What types of meetings do you usually host/participate in? Planning Is Selected

If you are participating in a planning meeting, how often is it clear what work has been assigned to you?

- ○ Always
- ○ Most of the time
- ○ About half the time
- ○ Sometimes
- ○ Never

+ Add page break

Q10

As a meeting host, would you want to receive additional feedback from each team member/participant after the meeting?

- ○ Yes
- ○ No

Q11

How often do your meetings go overtime?

- ○ Most of the time
- ○ Sometimes
- ○ Rarely
- ○ Never

Q13

Would you want your meeting platform to display remaining meeting time?

- ○ Yes
- ○ No

Q12

Does your team use additional tools for documenting meeting results?

- ○ Yes
- ○ No

Q21

Display this question

If Does your team use additional tools for documenting meeting results? Yes Is Selected

Do you consider such tools effective?

- Yes
- No
- I'm not sure

Q14

Do you ever have spectators in your meetings or do you ever participate in meetings as a spectator?

- Yes
- No

+ Add page break

Q15

Would you want to be able to host polls in your meetings?

- Yes
- Maybe
- No

Q16

Would you want to be able to import your documented meeting results into other platforms?

- Yes
- Maybe
- No

Q17

Display this question

If Would you want to be able to import your documented meeting results into other platforms? Yes Is Selected

Or Would you want to be able to import your documented meeting results into other platforms? Maybe Is Selected

Which platforms would you want your meeting software to integrate with?

- [ ] Project Management software
- [ ] Meeting hosting software
- [ ] Communication software
- [ ] Custom system
- [ ] Other
- [ ] All of the above

Q22

Please add any additional comments