

[Requirements] Elicitation

“Develop a series of questions (at least 4) to ask these target users to understand what problems they might have with teamwork-driven communication tools that are currently unsolved by Beans. Give these questions to your target users and record their answers.”

Question 1:

What customisation features can we add to ensure a Beans user experience that is more tailored to your preferences?

Question 2:

Did you feel restricted when using the current communication features and which communication features would you like Beans to add to ensure better user satisfaction?

Question 3:

In regards to using the channels, would you like to see any additions here to enable easier working for tasks?

Question 4:

Have you faced any challenges regarding the app not working and was there an easy way to know if it was from your end or ours?

Suggest running a backup server where the app responds by saying the app isn't working right now and stuff.

Question 5:

Which general improvements would you like Beans to implement to make sure the user experience is better?

Answers from target users:

Name: Lakshaya Kaushik

Email: lakshayakaushik35@gmail.com

Question 1: “I would like to see a few different colour themes for the software.”

Question 2: “I would like Beans to have a voice call functionality as I’m more motivated to do work when I’m on a call with my friends.”

Question 3: “I would like to see a files subsection in the channels so I can upload and view relevant files.”

Question 4: “There have been quite a few times when Beans was not working and it would take considerable time to figure out whether the problem was on my PC or the software was down.”

Question 5: "There should be a calendar feature to view any upcoming deadlines."

Name: Rohan Agarwal

Email: rohanagarwal600@gmail.com

Question 1: "I would like to have a soccer wallpaper on the frame over which the other functionalities can exist."

Question 2: "I would like Beans to have a video call and streaming functionality as I'm able to share what I'm working on live."

Question 3: "I would like to see which members of a specific channel are online and also show online statuses for all users."

Question 4: "The software has not yet crashed for me and works fine."

Question 5: "There should be a feature where I'm able to send images, videos or gifs to the members in a channel."

Once you have done this, think about how you would solve the target users' problem(s) and write down a brief description of a proposed solution.

1. Problem: No colour themes are available for the software.
Solution: Create multiple themes with a mix and match of different colours that apply to the whole software.
2. Problem: There is no voice call function.
Solution: The software must implement a voice call function that the users can use in the channels and dms they are part of.
3. Problem: There is no 'files' subsection in the user channels.
Solution: A files subsection must be created so the users can upload, download and view files and collectively collaborate on team projects.
4. Problem: The users are not able to make out whether the software servers are down.
Solution: A backup server must be created so it intercepts and produces an error message informing the user that the software is down.
5. Problem: There is no calendar feature on Beans.
Solution: A calendar subsection should be created for users where they can create deadlines for themselves.
6. Problem: There is no feature to add a wallpaper for the software.
Solution: The software should allow users to add a wallpaper that flows through the whole software and other functionalities are added on top of that wallpaper. The users would be allowed to upload their own images or choose from some basic ones provided.
7. Problem: Beans does not have a video call and streaming feature.
Solution: New functionality of video calling and streaming should be added so users are able to video call other users in the channels and dms they are part of.

8. Problem: No online status feature is available for each user.
Solution: A new function must be added where the online status exists and should be added next to the user handle.
9. Problem: The channel does not indicate which users within it are online currently.
Solution: A list should be added to the channels that show all the users that are online.
10. Problem: The user is restricted to only sending text-format messages as of now.
Solution: Beans must add the functionality where users are able to send messages in video, image and gif formats.

[Requirements] Analysis & Specification - Use Cases

Take the responses from the elicitation step and express these requirements as user stories (at least 3). Document these user stories. For each user story, add user acceptance criteria as notes so that you have a clear definition of when a story has been completed.

	User Story	Acceptance Criteria
1.	As a student, I want to work on group projects with my team. Meeting up physically at one place at a specific time is not always feasible and online is more flexible. Therefore, the video call feature with a streaming option is crucial for times like these.	<ul style="list-style-type: none"> - Dms and channels will both provide the functionality to call other users in the respective dm or channel. - A button with a video icon must be added in the top right corner which will signify a video call option. - Any valid user of that channel or dm can initiate a call as soon as this button is clicked. - The user will enter the call with their camera turned on and muted by default and once in the call, they will be able to see the streaming feature which will be signified by a mini-screen icon button. - The maximum capacity of a call will be the number of members in that specific channel or dm.
2.	As a student, I have a lot of deadlines for various assignments and tests. As of now, the information shared through channels and dms about deadlines has to be manually entered into a different app. Therefore, a feature that enables me to save the deadline details on the calendar in the Beans app will allow me to be more	<ul style="list-style-type: none"> - A calendar subsection must be added to the software in the left tab where a calendar exists that is synced with the current date, time and year. - A user is able to edit this calendar by clicking on specific sections of the calendar and entering the deadline details and hitting save. - This will create a new event on the calendar where the user is able to view the deadline successfully. - Further, the teachers will also be able to access this calendar and can create an event deadline by making it on their calendar and forwarding

	organise it as all the information will be on one software.	this event to the whole class who will be able to view this information on their calendars in the Beans software.
3.	As a teacher, when I enter a channel, I must be able to see a list of all the students that are online which will make role marking easier.	<ul style="list-style-type: none"> - The activity status must be created which will be a green, yellow or red dot next to the user's handle. - If the user is on the app and engaged in some activity, the activity status will automatically be green, if the user is idle on the app, the status will be yellow and if they are not online, it will be red. - In a channel, there must be a list of all the users that will be displayed on the left and everyone will be able to see each other's activity status.

USE CASE:

Calendar feature -

1. The user clicks on the calendar subsection on the front end.
2. Front end forwards this request to the backend.
3. The backend checks whether the user is authorized, if yes, it allows the user to access the calendar.
4. This data is forwarded to the front end, where the user is able to view the calendar.
5. The user hovers over specific data and time on the front end and clicks on the add event button.
6. The details entered by the user are sent to the backend to process.
7. The backend approves the data and relays the confirmation back to the front end where the user is able to view the desired deadline set correctly on the calendar.
8. The user after successfully setting a deadline, clicks on the deadline and clicks the add reminder button.
9. This information is sent back to the back end for approval.
10. The back end confirms it and sends it to the front end where the user sees a bell icon next to the deadline.

USE CASE:

Video call with streaming feature -

1. The user clicks on a channel they wish to join on the front end.
2. Front end forwards this request to the backend.
3. The backend checks whether the user is allowed to join that specific channel.
4. Once the approval is sent by the backend, the user enters the channel on the front end.
5. To initiate a video call, the user clicks on the button with the video icon on the front end.
6. The backend checks whether the user is valid and has the right to initiate a call. Once authorised the user enters the call.
7. Back end forwards the data regarding the call features to the front end.
8. The front end displays the buttons of unmute/mute, streaming and leaving the call.

9. The user clicks on the icon he wants to use, which triggers a data request to the backend.
10. The backend approves the desired feature and forwards the data to the front end to display.
11. One of the users clicks the 'streaming' button.
12. This data is forwarded to the back end for approval.
13. The back end approves the feature and relays the data back to the front end.
14. The front end uses this data to display the call with the screen details of the user that triggered the stream button.
15. One of the users in the call clicks on the leave call button.
16. The front end sends this data to the back end for approval.
17. The back end checks if the user is currently participating in the call and if correct, removes the user from the call.
18. This data is sent back to the front end where the user is back on the channel page and not the call page.
19. After all the users have requested to leave, the backend checks if the call has no participants left and ends the call.
20. This is forwarded to the front end where the call does not exist anymore and there are no call details being displayed on the channels page.

[Requirements] Validation

With your completed use case work, reach out to the 2-3 people you interviewed originally and inquire as to the extent to which these use cases would adequately describe the problem they're trying to solve. Ask them for a comment on this, and record their comments in the PDF.

Lakshaya Kaushik (lakshayakaushik35@gmail.com):

"The calendar feature will enable me to plan ahead and check the deadlines daily."

Rohan Agarwal (rohanagarwal600@gmail.com):

"The video call with the streaming feature will now make working on group projects very easy and effective."

[Design] Interface Design

Now that we've established our problem (described as requirements), it's time to think about our solution in terms of what capabilities would be necessary. You will specify these capabilities as HTTP endpoints, similar to what is described in 6.2. There is no minimum or maximum of what is needed - it will depend on what problem you're solving.

<u>Name & Description</u>	<u>HTTP Method</u>	<u>Data Types</u>	<u>Exceptions</u>
<p>channel/statuslist/v1</p> <p>When a valid user enters the channel, they are able to view a list that displays all the members of the channel and their activity status.</p>	GET	<p>Parameters: (channelId)</p> <p>Return type if no error: { }</p>	<p>400 Error when:</p> <ul style="list-style-type: none"> channelId does not refer to a valid channel <p>403 Error when:</p> <ul style="list-style-type: none"> channelId is valid and the authorised user is not a member of the channel. Token is invalid
<p>calendar/join/v1</p> <p>When an authorised user enters their calendar subsection, they are able to view a calendar.</p>	POST	<p>Parameters: (token)</p> <p>Return type if no error: { }</p>	<p>403 Error when:</p> <ul style="list-style-type: none"> Token is invalid
<p>calendar/create/deadline/v1</p> <p>When a valid user creates a deadline, they are able to view it on their calendar.</p>	POST	<p>Parameters: (token)</p> <p>Return type if no error: { }</p>	<p>400 Error when:</p> <ul style="list-style-type: none"> Start time is after the end time. The date and time being requested are in the past. <p>403 Error when:</p> <ul style="list-style-type: none"> Token is invalid
<p>calendar/create/reminder/v1</p> <p>When a valid user creates a reminder, they are able to view a bell icon next to their deadline which will remind them close to the deadline.</p>	POST	<p>Parameters: (token)</p> <p>Return type if no error: { }</p>	<p>400 Error when:</p> <ul style="list-style-type: none"> Start time is after the end time. The date and time being requested are in the past. <p>403 Error when:</p> <ul style="list-style-type: none"> Token is invalid
calendar/saveandexit/v	POST	Parameters:	403 Error when:

1 The changes made to the calendar are saved and the user is returned to the home page.		(token) Return type if no error: { }	<ul style="list-style-type: none"> Token is invalid
channel/join/videocall/v1 When a valid user clicks on the video call button, they enter a video call.	POST	Parameters: (channelId) Return type if no error: { }	400 Error when: <ul style="list-style-type: none"> channelId does not refer to a valid channel 403 Error when: <ul style="list-style-type: none"> channelId is valid and the authorised user is not a member of the channel. Token is invalid
channel/leave/videocal l/v1 When a valid user clicks on the leave video call button, they return to the page of the main channel.	POST	Parameters: (channelId) Return type if no error: { }	400 Error when: <ul style="list-style-type: none"> channelId does not refer to a valid channel 403 Error when: <ul style="list-style-type: none"> channelId is valid and the authorised user is not a member of the channel. Token is invalid
channel/stream/v1 When a valid user clicks on the stream button, their screen is shared on the call screen and all the users in the call will be able to view it.	POST	Parameters: (channelId) Return type if no error: { }	400 Error when: <ul style="list-style-type: none"> channelId does not refer to a valid channel 403 Error when: <ul style="list-style-type: none"> channelId is valid and the authorised user is not a member of the channel.

			<ul style="list-style-type: none">• Token is invalid
--	--	--	--

[Design] Conceptual Modelling - State Diagrams

Now that you have a sense of the problem to solve, and what capabilities you will need to provide to solve it, add at least ONE state diagram to your PDF to show how the state of the application would change based on user actions. The aim of this diagram is to help a developer understand the different states of the application.

