MEETING MINUTES

Date	Tuesday 5/4/22
Time	8-8:30pm
Attendees	Tom, Calvin, Daniel, Eileen, Aayush
Apologies	
Please bring/read	Iteration 3 spec + hayden's video
Meeting purpose	Iteration 3 task delegation
Next meeting	Thursday lab time Then Thursday 9pm - discuss next steps

Agenda

Agenda Item	Minutes
1. Update on Iteration 2 failed tests	FIxed all issues except 1: - the last test - Tom and Daniel to merge their fixes tonight - message on Messenger when need us to accept merges - Record of fixed tests: Iteration 2 checklist
Second weekly standup scheduled	Thursday nights - 15 minutes 9pm
New http endpoints delegation	notifications/get/v1 - tom search/v1 - daniel message/share/v1 - tom message/react/v1 - daniel message/unreact/v1 - daniel message/pin/v1 - Calvin message/unpin/v1 - Calvin message/sendlater/v1 - Eileen message/sendlaterdm/v1 - Eileen standup/start/v1 - Tom standup/active/v1 - Tom + Daniel standup/send/v1 - Tom + Daniel auth/passwordreset/request/v1 - Tom auth/passwordreset/reset/v1 - Aj user/profile/uploadphoto/v1 - Calvin user/stats/v1 - Eileen users/stats/v1 - Eileen Deployment - Aayush

	notifications/get/v1 — Thomas search/v1 — Daniel message/share/v1 — Thomas message/react/v1 — Daniel message/unreact/v1 — Daniel message/pin/v1 — Calvin message/unpin/v1 — Calvin message/sendlater/v1 — Eileen message/sendlaterdm/v1 — Eileen standup/start/v1 — Daniel/Thomas standup/active/v1 — Daniel/Thomas standup/send/v1 — Daniel/Thomas auth/passwordreset/request/v1 — Thomas auth/passwordreset/reset/v1 — Aayush user/profile/uploadphoto/v1 — Thomas user/stats/v1 — Eileen users/stats/v1 — Eileen Deployment — Aayush
Iteration 3 Tasks	- **Write up your own task issue boards.** - **Understand the lecture content from Week 7** - **Keep working on your previous aspects:** - **E.g. if you did messages, keep working on refining functions on messages** - **Making the code more pythonic** - **Keep excess code in helper functions** **New tasks:** 1. **Document the planning of new features.**

- **You are required to scope out 2-3 problems to solve for future iterations of Seams. You aren't required to build/code them, but you are required to go through SDLC steps of requirements analysis, conceptual modelling, and design.**
- **Full detail of this can be found in 5.6.**

2. **Deploy your backend to the cloud.**
   - **You are required to deploy your backend a cloud-provider so that it can be accessed from anywhere in the world.**
   - **Full detail of this can be found in 5.7**

5.6. Planning for the next problems to solve

Software development is an iterative process - we're never truly finished. As we complete the development and testing of one feature, we're often then trying to understand the requirements and needs of our users to design the next set of features in our product.

For iteration 3 you are going to produce a short report in planning.pdf and place it in the repository. The contents of this report will be a simplified approach to understanding user problems, developing requirements, and doing some early designs.

N.B. If you don't know how to produce a PDF, you can easily make one in Google docs and then export to PDF.

We have opted not to provide you with a sample structure - because we're not interested in any rigid structure. Structure it however you best see fit, as we will be marking content.

[Requirements] Elicitation

[Requirements] Elicitation Interview Questions

Find 2-3 people to interview as target users. Target users are people who currently use a tool like Seams, or intend to. Collect their name and email address.

Develop a series of questions to ask these target users to understand what *problems* they might have with teamwork-driven communication tools that are currently unsolved by Seams. Give these questions to your target users and record their answers.

Once you have done this, think about how you would solve the following problem and write down a brief description of a proposed solution.

[Requirements] Analysis & Specification - Use Cases

Once you've elicited this information, it's time to consolidate it.

Take the responses from the elicitation and express these requirements as user stories. 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. **Once documented, generate at least one use case that attempts to describe a solution that satisfies some of or all the requirements elicited. You can generate a visual diagram or a more written-recipe style, as per lectures.** ### [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.** ### [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.** ### [Design] Conceptual Modelling (State) **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.** ## 5.7. Deployment **You and your team are to host your backend on a cloud-provider. Once your backend has been deployed to the cloud, you will be able to point the frontend to use the new URL of where the backend is deployed and use your backend from anywhere in the world. In summary:** - **You get your server (that you wrote) deployed to the internet at a public URL** - **You still run your frontend locally (which can connect to that server)** **Depending on how you and your team have structured your project, your current method of using data may have to be rethought. Deploying to cloud and developing locally require two different mindsets and you and your team may find that you held some assumptions that are valid when developing locally but do not hold when being hosted on the cloud.** **We have written a guide on how to deploy to a free cloud provider AlwaysData. Click here to view the guide.**
Bonus Marks	**Language filter - filter out all swear words**

	Auto-doc strings - add this extension to VSCode to auto-write your docstrings for you 

Action Items

Person	Action Item	Due Date
Everyone	1. Finish 1 function each by this Thursday - lab time 2. Finish all your delegated functions & deployment by Monday - for first leaderboard run 3. Add your functions to the Issue board 4. Requirements Elicitation interview: 1 team-member interview 1 person each Put your interview questions in this doc: [Requirements] Elicitation Interview Questions Reminders: - Keep the repo clean - 1 branch per function and its tests - Write up your own task issue boards. - Understand the lecture content from Week 7	1 function done **By Thursday lab time** Get all http endpoints implementation done **by Monday** (first leaderboard run)
Tom	notifications/get/v1 - tom message/share/v1 - tom standup/start/v1 - Tom + Daniel standup/active/v1 - Tom + Daniel standup/send/v1 - Tom + Daniel auth/passwordreset/request/v1 - Tom	
Calvin	message/pin/v1 - Calvin message/unpin/v1 - Calvin user/profile/uploadphoto/v1 - Calvin	
Daniel	message/react/v1 - daniel	

	message/unreact/v1 - daniel standup/start/v1 - Tom + Daniel standup/active/v1 - Tom + Daniel standup/send/v1 - Tom + Daniel	
Eileen	Email Usmaan: - Meeting minutes from today - Welcome to join Thursday lab or 9pm standup (15 mins) HTTP endpoints: message/sendlater/v1 - Eileen message/sendlaterdm/v1 - Eileen user/stats/v1 - Eileen users/stats/v1 - Eileen	
Aayush	auth/passwordreset/reset/v1 - Aj Deployment - Aayush Look at **Week 8 Lecture 1** for **deployment**	