**Preparation (5 minutes)**

* Each team divide into two groups:
  + 2 users -- generalist, specialist.
    - Eashan – real estate agent
    - Wilmer - generalist
  + 3 developers -- Data, Code, Interface.
    - Aveline - NLP D
    - Victor – Code and interface
* Use a **Google Doc** (or something similar) to record your work.
  + Or, use another word process and designate a **scribe.**

**Step 1: Define the scope of your project (5 minutes)**

* What is main purpose of the project?

Real Estate Agent Customer Support Assistant Chatbot

* What is the value proposition?

No need for extra assistants, 24/7 on call, chatbot could text clients directly about appointments, forms to be completed before appointment, concerns, and ETA response time.

* Not a mission statement or vision.
* Designed to allow you to accept or reject features and designs.

**Step 2: Elicit user stories (10 minutes)**

* As a real estate specialist, I would like my lead spreadsheet to be ingested by the software via upload or cloud storage and for it to draft emails to follow-up with leads with availability.
* The software should recognize changes to my calendar and be synced with it.
* The software should list the names of leads, along with their contact information
* If it could include the status of the customer on the marketing/sales funnel
* It should notify me when new messages / touches have arrived and meetings have been scheduled
* It should include shortcuts with stock/pre-formulated responses, that can be timed, sent automatically, or manually
* A real estate specialist should be able to access this on the desktop, mobile app, and cloud.
* It should work across multiple OSs and calendar types (Apple, Microsoft, Google, ICQ, etc )
* Should have a search feature for leads with filters by status, date, etc.

**Step 3: Convert user stories into requirements (10 minutes)**

* Functional: Required feature-read in lead csv files and reach out to potential clients via text/email
* Functional: Ability to read calendar software and make live changes based on calendar status
* Functional: Include search feature for leads with various filters
* Non-functional: The system should refresh calendar status every 5 minutes automatically and instantly when user interacts with software
* Non-functional: The system should be able to handle 50,000 users without performance deterioration
* Think of a requirement as a feature – what the product will do.
* Think of functional requirements as guides to actually develop code.
* Create so-called non-functional requirements, too.

**Step 4: Prioritize requirements (10 minutes)**

* Prioritize requirements by giving them a status of M, S, C, W
* Must Haves: Automated reaching out to potential clients, automated scheduling feature, ability to read calendar software and modify calendar,
* Should Haves: home budget calculator with financial information collected from customer directly via email/text
* Could Have: possible e-signature of real estate forms, collect payments via portal
* Won’t Have:
* Throw out everything but the Ms

**Step 5: Show and tell**