Chatbot

input : participant . xlsx

wechat_id	treatment

load into ↓ database

user
user_id
day ← refers to last completed starting from 0
wechat_id
treatment

① 6pm, 10pm
paste "/completion" page info into column in spreadsheet

② Run macro to get "day" column in spreadsheet to .csv for chatbot

③ Run chatbot.py
login through scanning uuID
send text + link to each

↑ chatbot uses same hashing function

refers to next to be send, starting from 1

user_id	day	wechat_id	treatment	(continued) →

←	day 1 - 6pm	day 1 - 10pm	day 2 - 6pm	day 2 - 10pm	day 3 - 6pm	...
	0	1	send 2			
	0	0				
	0	0	1	send 2		
	0	0	0			
	0	0	0	0		

same_day reminder

next_day reminder

drop

completion

completion → day

denotes drop

Excel Macro : if NOT (-1) then
day = max of completion + 1

↑ ②

① completion columns obtained from
<url>/completion : webpage displaying max completion day from database, user table, ranked ASC by user_id

chatbot is containerized so that it doesn't have to keep running.

prompt or reminder

info Page

url

/<hashid-user_id>/<hashid-day>/info

↓ decode

user_id day

treatments
- treatment
- day
- event_id

treatment from treatments table join user table in database

display infopage

④

? Randomize

TITLE
SUBTITLE
TIME
LOCATION

IMAGE

...

infopage blueprint + content (identified by event_id)

survey ⑤

NEXT

user_id
day,

Display survey page
render_template('survey <day>.html'
user = user)

[user_id, day]

Hash user_id, day by

redirecting from /<user_id>/<day>/survey

⇒ /<hashid_user_id>/<hashid---> /survey

⑥

question

NEXT

SUBMIT

[question_id]
(TEXT)

[survey_id]
(INT primary key)

⇒ saved to survey table in database with [timestamp]

when last "Submit" button is clicked
save in activity table in database
with [day], [current url] ...

↓
update [day] in user table

⑦ display complete & thanks page → completion refleted in

<url>/completion

<url>/completion/details

→ chatbot back to ①