

## **15112 Design Document**

### **Project Proposal:**

My proposal is to build an interview bot to replace a human interviewer. This will automate the interviewing process and, also provide an opportunity to take an unbiased interview and determine how much of a match the interviewee is to the people currently in the position they are getting interviewed for.

The modules that I'm using are: Watson API: Personality Insights, Speech to Text, Text to Speech. And Django for the user interface. The main ideas which I will code aside from the linking between files is the LinkedIn crawler which will give me data about the people in the position the interviewee is applying for and thus the data against which I will compare the interviewee and also how I'm going to compare the sets of data, which abilities to give preference to (something that will be given to the program when initializing the interview).

The main competitor, in the functional sense, in this area is the Mya interviewer AI, which as of now automates 75% of the interview process. It is a chatbot, the interview occurring through messages instead of a face-to-face interview. Mya can ask

- 1) simple pre-screening questions,
- 2) answer frequently asked questions,
- 3) provide updates on the progress of your application,
- 4) offer tips and guidance to candidates, and
- 5) administer assessments and challenges.

It provides analysis on how qualified and engaged each interviewee is and how likely is their acceptance of an offer, along with good consumer experience. Also, Mya is intelligent in that she tailors her questions per the interviewee responses to previous questions. The interviewee can ask for clarification anytime during the interview. In the case where Mya cannot answer a candidate's question, she forwards that question to the human recruiters, while storing the answer in her memory, who also can oversee and assist when required.

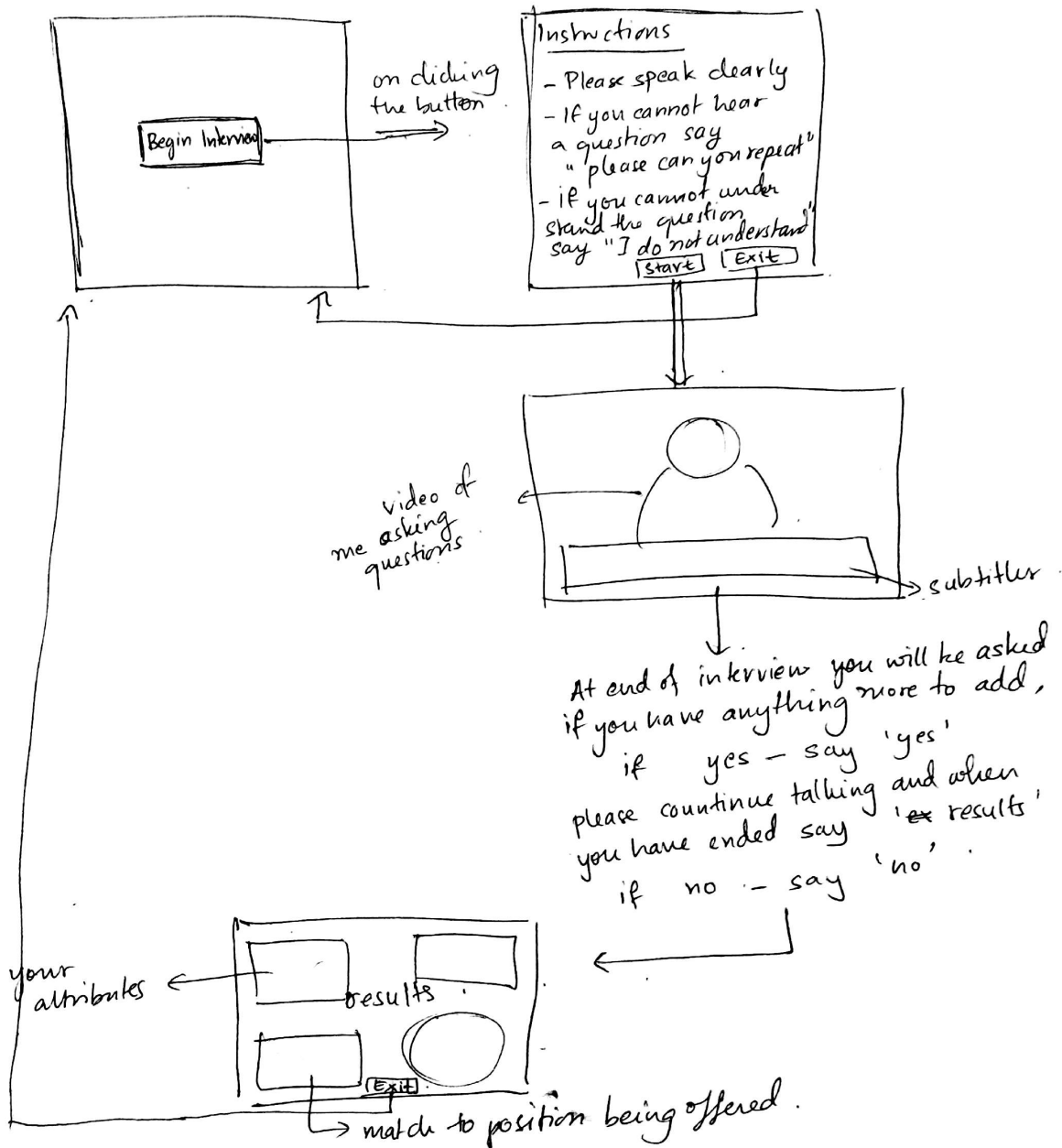
I do not particularly like that it is a chatbot as I've experienced myself that I think thrice before actually sending anything which might be an advantage in certain cases i.e. you think about what you are saying however that also takes out the human interaction and makes the interviewee distant, so what I'm doing is having speech to make it more realistic having a video of me, asking standard questions and a computer generated voice to have actual data which is easier to understand and analyze as most recruiters are used to analyzing spoken interviews. To ask for clarification the interviewee has to say a pre-defined phrase which will cause the program to repeat in a louder volume or if specified reframe the question.

Also, to build the comparison to the requirements of the position, I'm building something like a LinkedIn crawler to find people with the same position and compare their personality to you and see the match.

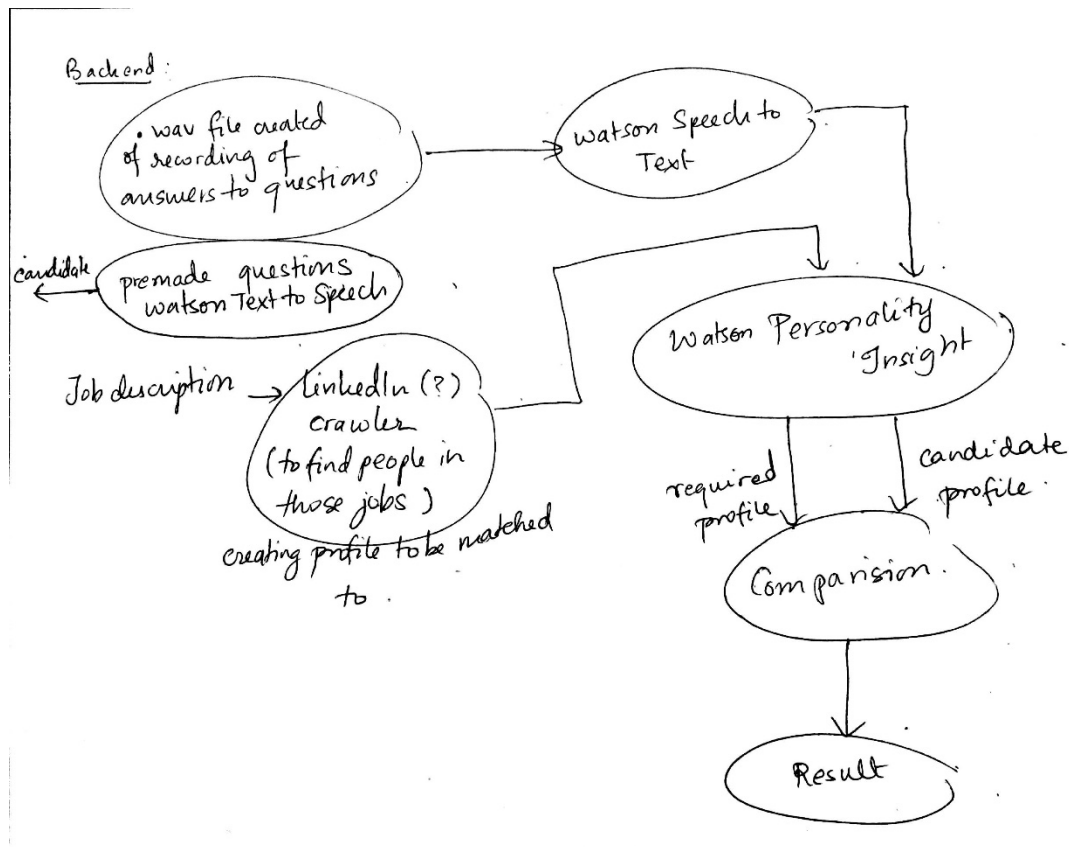
### **User Interface Comments :**

Background pictures and effort to bring an office environment and the interviewer picture especially to bring reality to the program.

Speech-to-Text and Text-to-Speech help in having natural flow of communication.



Storyboard : User Interface:



There will be a separate page to input the variables i.e job-title and which abilities to prefer over others.

### Design Features :

#### Things that can be added :

- IBM Watson Tone Analyzer that would give us a more accurate understanding of the candidate
- More data collection for jobs to have a more robust comparison data. I went in breadth, next step is to go in depth.
- CGI generated Interviewer/ Video of recorded speech
- Highlight personality traits which exceed the threshold comparison data.

#### Things that I did :

- Matching personality to other jobs (Are you a better match to other jobs rather than the one you are applying for ?)
- Changed Text to Speech, and Speech to Text of Watson to Google, as more support and documentation for python and quicker as doesn't require an API request and call to the website
- Changed data source from Linked In Profile Summaries to Cover Letters : increased number of jobs that can be handled by the bot (761 jobs)

- Changed User Interface from Django/WebApps to Tkinter - easier integration with the modules plus saved time that would have been spent in learning to link the python modules to html,css,and js. Through Tkinter the whole User Interface is python based.

-Split result page into two, in depth personality analysis and job match.

-Admin page to select job for which we are interviewing

-Images for background of pages + buttons for navigation