



# **Project Proposal**

Open source based Learning App

Group 9

07/10/2019



## ***Part 1.Features analysis***

### **The login part of application**

Under the consideration that we are going to have two kinds of users- mentors and learners, we plan to implement “one entrance to different users” principle. This means the mentors and learners will have the same registration and login interface and what different is that user must choose their attribute (mentor or learner) at the beginning of registration and login.

Use case: register and login

User	All user
Work flow	User register a account for themselves. When user doing registration, user have to choose either to be a learner or a mentor
Precondition	-
Result	Different permission for user

### **Main functions for learner**

After learner logged into the extension, learner can start asking their question about coding when they are browsing Github. The extension will provide three main functions for learner: asking question by keyword; asking explanation for a whole sentence of coding; asking causal problem.

#### **(1) search by keywords**

The learner can search keywords of program, like for, while, if .... Then the chatbot will give the explanation of this keywords and tell how to use these kinds of keywords.

Use case: asking explanation of key words

User	Learner
Work flow	The extension will show as a chatroom, all the user has to do is inputting the correct keywords they don't understand and click the send button. Then just wait a few second (less than 10 seconds) the chatbot will give back the explanation and how to use the keywords
Precondition	User has to login first
Result	Give explanation of the keywords and tell learner how use the keywords

#### **(2) search one completed sentence of program**

The learner can ask question by input a whole (completed) sentence of program. Then the chatbot will give explanation of this sentence

Use case: asking explanation of sentence

User	Learner
Work flow	User needs to input the sentence which they don't understand and make sure the sentence is completed. Then wait for a while (less than one minute) to get the answer.
Precondition	User has to login first
Result	If the chatbot can give the explanation it will show directly or the chatbot will suggest user to transfer channel to the manual response.

#### **(3) search for general question**

The learner can ask question not related to coding, just daily greeting or normal communication

Use case: daily greeting

User	Learner
Work flow	User click the extension and input normal question or greeting words, then the chatbot can do normal communication with user.
Precondition	User has to login first

Result	Normal chatting.
--------	------------------

### **Main function for mentors**

The mentor can answer the question, which cannot be explained by chatbot, and communicate with learner interactively.

Use case: explain question manually

User	Mentor
Work flow	Mentor wait for connection from learner and accept it. Then start answer the question raised by the learner
Precondition	User has to login as a mentor
Result	Mentor teach learner how to do coding and problem get solved

### **Main function for information collection part**

During the process that question get solved by mentors, the application will collect the question raised more frequently (top 5). Then save the correct answer given by mentor into database of chatbot.

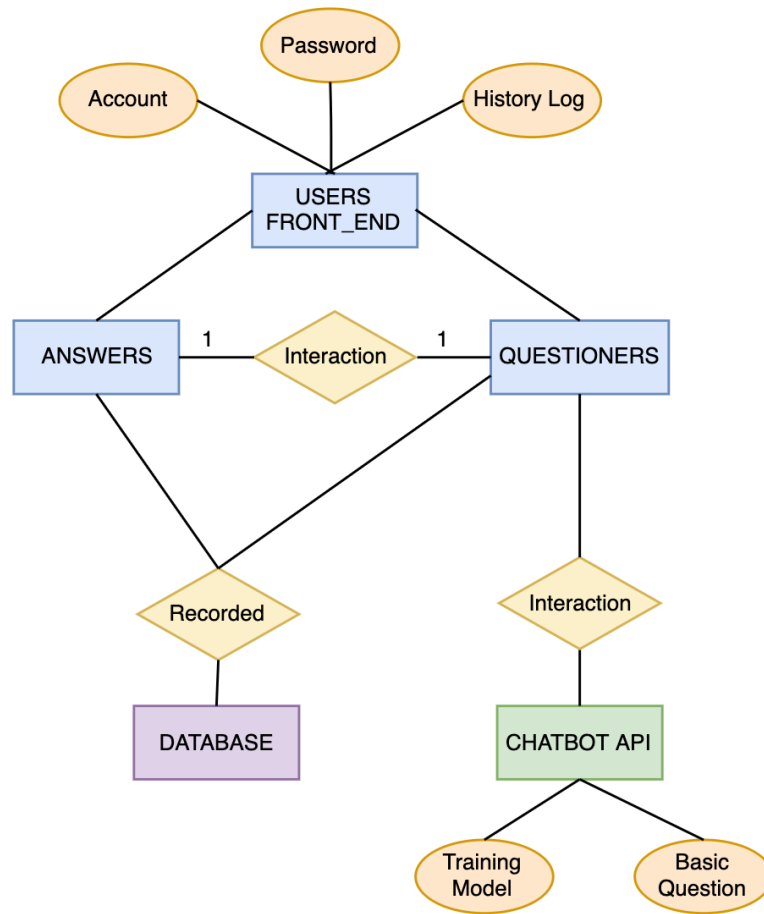
Use case: data collection

User	Training center
Work flow	The application will collect and record the question answered by mentor and save the record into the database. Then we will choose top 5 most question solved by mentors and store the question in the database of chatbot.
Precondition	It needs a period of accumulation
Result	Next time when the same question raised by learner it won't need to wait the mentor to answer it. The chatbot can solve it automatically.

## ***Part 2. System Design***

The main elements of our application includes 4 parts. The first part is USER FRONT END, which can split into Answers or Questioners according to the role of users. There are three attributes in this entity; account and password are used for login, and users' history log. The second part is BACK END API, this part is used for control most of the data flow during the interaction between the questioners and answers and achieve most of functions. The third part is CHATBOT API, it offers basic interaction and gives response for simple questions, and it can train some questions and giving feedback. The last part is DATABASE which is for recoding data.

- (i) An Entity-relationship (ER Model) of data layer.



(ii) The System architectural diagram below, showing the main system components and interactions of our application.

